

Befehlsübersicht

Die folgende Liste enthält die im Compiler integrierten Befehle, unterteilt in 21 Gruppen der verwendeten Befehle.

Es kann dabei helfen, den besten Befehl für die jeweilige Aufgabe zu finden.

Einige Befehle können in mehreren Gruppen vorkommen.

Befehls-Kategorie	Befehls-Nr	Seite
1. Array-Operationen	(1 - 53)	2
2. Collection Objekte	(54 - 93)	5
3. COM Operationen	(94 - 136)	8
4. Kommunikations-Steuerung	(137 - 164)	10
5. Compiler Operationen	(165 - 214)	11
6. Debugging und Fehlerkontrolle	(215 - 241)	13
7. Befehle der dynamischen Dialogtools	(242 - 520)	14
8. Dateibefehle	(521 - 569)	25
9. Ablaufsteuerung	(570 - 624)	27
10. Grafikkommandos	(625 - 762)	29
11. Eingabekommandos	(763 - 790)	34
12. Speicherverwaltung	(791 - 829)	35
13. Meta-Anweisungen	(830 - 858)	37
14. Numerische Operationen	(859 - 968)	38
15. Betriebssystem	(969 - 1024)	43
16. Druckbefehle	(1025 - 1150)	45
17. String-Operationen	(1151 - 1321)	49
18. Textbefehle	(1322 - 1331)	56
19. Thread-Steuerung	(1332 - 1362)	57
20. Zeitbefehle	(1363 - 1405)	58
21. Sonstige Operationen	(1406 - 1412)	60

1. Array-Operationen

Die folgenden Funktionen können zur Bearbeitung und Verwaltung von Arrays verwendet werden:

LfdNr	Befehl	Beschreibung	Seite
1	#DEBUG ERROR	Steuerung der Generierung des Fehlerprüfcodes	xxx
2	#DIM	Gibt an, ob Variablen vor der Verwendung deklariert werden müssen	xxx
3	ARRAY ASSIGN	Zuweisung einer Reihe von Werten aufeinander folgenden Elemente eines Arrays	xxx
4	ARRAY DELETE	ein einzelnes Element aus einem bestimmten Array löschen	xxx
5	ARRAY INSERT	ein einzelnes Element in ein gegebenes Array hinzufügen	xxx
6	ARRAY SCAN	Scannen (durchsuchen) das gesamte oder einen Teil eines Arrays nach einem bestimmten Wert	xxx
7	ARRAY SORT	Sortieren des gesamten oder einen Teil eines gegebenen Arrays	xxx
8	ARRAYATTR	Gibt beschreibende Attribute eines gegebenen Arrays zurück	xxx
9	BIT CALC	Setzen oder Rücksetzen eines Bits in einem impliziten Bit-Array	xxx
10	BIT	Gibt den Wert eines bestimmten Bits in einem impliziten Bit-Array zurück	xxx
11	BIT	Einzelne Bits eines impliziten Bit-Arrays manipulieren	xxx
12	DATA	Deklariert ein Array von Konstanten, die von READ\$ gelesen werden sollen	xxx
13	DATACOUNT	Gibt die Gesamtzahl der lokalen Datenelemente zurück	xxx
14	DIM	Deklariert und Dimensioniert von Arrays, Skalarvariablen und Zeigern	xxx
15	ERASE	Array-Speicher freigeben	xxx
16	FILESCAN	eine geöffnete Datei schnell scannen, bevor mit GET in ein Array geladen wird	xxx
17	GET	ein vollständiges Array aus einer Binärdatei einlesen	xxx
18	IPowerArray.ARRAYBASE	Gibt die Adresse des ersten Elements des Arrays zurück.	xxx
19	IPowerArray.ARRAYDESC	Gibt die Adresse des SAFEARRAY-Deskriptors zurück	xxx
20	IPowerArray.ARRAYINFO<Get>	Ruft die Infozeichenfolge ab, sofern eine vorhanden ist.	xxx
21	IPowerArray.ARRAYINFO<Set>	weist den Infostring zu.	xxx
22	IPowerArray.CLONE	Es wird ein exaktes Duplikat des SafeArrays erstellt und im angegebenen PowerArray-Objekt gespeichert.	xxx

23	IPowerArray.COPYFROMVARIANT	Es wird eine exakte Kopie des angegebenen SafeArrays erstellt und in diesem PowerArray-Objekt gespeichert	XXX
24	IPowerArray.COPYTOVARIANT	In diesem Objekt wird eine exakte Kopie des SafeArrays erstellt und im spezifizierten Variant gespeichert	XXX
25	IPowerArray.DIM	Dimensioniert (erstellt) ein neues Array.	XXX
26	IPowerArray.ELEMENTPTR	Ruft die Adresse des angegebenen Datenelements ab.	XXX
27	IPowerArray.ELEMENTSIZE	Ruft die Speichergröße (in Bytes) jedes Datenelements des Arrays ab.	XXX
28	IPowerArray.ERASE	Zerstört das enthaltene Array und leert das Objekt.	XXX
29	IPowerArray.LBOUND	Ruft die untere Grenzzahl für die angegebene Dimension ab.	XXX
30	IPowerArray.LOCK	Erhöht die Sperranzahl des SAFEARRAY.	XXX
31	IPowerArray.MOVEFROMVARIANT	Überträgt den Besitz des angegebenen SafeArrays auf das PowerArray-Objekt.	XXX
32	IPowerArray.MOVETOVARIANT	Überträgt den Besitz des in diesem PowerArray-Objekt enthaltenen SafeArrays auf einen Variant-Parameter	XXX
33	IPowerArray.REDIM	Ermöglicht das Löschen und Neudimensionieren des SafeArrays auf eine neue Größe.	XXX
34	IPowerArray.REDIMPRESERVE	Ermöglicht die Änderung der niedrigstwertigen (äußersten) Grenze auf eine neue Größe. Die verbleibenden Datenelemente im Array bleiben erhalten.	XXX
35	IPowerArray.RESET	Alle Elemente im SafeArray werden auf ihren ursprünglichen Standardwert zurückgesetzt.	XXX
36	IPowerArray.SUBSCRIPTS	Ruft die Anzahl der Dimensionen (Indizes) für dieses Array ab.	XXX
37	IPowerArray.UBOUND	Ruft die obere Grenzwertzahl für die angegebene Dimension ab.	XXX
38	IPowerArray.UNLOCK	Dekrementiert die Sperranzahl des SAFEARRAY.	XXX
39	IPowerArray.VALUEGET	Ruft den Wert des angegebenen Array-Elements ab.	XXX
40	IPowerArray.VALUESET	Weist dem angegebenen Array-Element den angegebenen Wert zu.	XXX
41	IPowerArray.VALUETYPE	Ruft den %VT-Code ab, der die in diesem Array enthaltenen Daten beschreibt.	XXX
42	JOIN\$	Gibt eine Zeichenfolge zurück, die aus allen Zeichenfolgen in einem Zeichenfolgenarray besteht	XXX
43	LBOUND	Gibt den niedrigsten Index der spezifischen Dimension eines Arrays zurück	XXX
44	LET	weist einem Array einen Variant oder einem Variant einem Array zu	XXX
45	LINE INPUT#	Liest Zeile(n) aus einer sequentiellen Datei in eine Zeichenfolgen-Variable oder ein Array	XXX
46	MAT	Matrixberechnungen auf numerischen Arrays	XXX

47	PARSE	Analysieren einer Zeichenfolge und extrahieren aller durch Trennzeichen getrennten Felder in einem Array	xxx
48	PRINT#	Schreibt ein vollständiges Array in eine sequentielle Datei	xxx
49	PUT	Schreibt ein vollständiges Array in eine Binärdatei	xxx
50	READ\$	Zeichenfolgendaten aus einer lokalen DATA-Liste abrufen	xxx
51	REDIM	Deklariert einen dynamischen Arrays, Speicher zuweisen, oder neu zuweisen und die Zuordnung freigeben	xxx
52	RESET	einen Array-Index setzen oder ein ganzes Array auf Null oder null/leer setzen	xxx
53	UBOUND	Gibt den höchsten Index der spezifischen Dimension eines Arrays zurück	xxx

2. Collection Objekte

Die folgenden Operationen bieten eine bequeme Möglichkeit, auf eine verwandte Gruppe von Elementen zu verweisen:

LfdNr	Befehl	Beschreibung	Seite
54	ILinkListCollection.ADD	am Ende der LinkListCollection wird ein Element hinzugefügt.	xxx
55	ILinkListCollection.CLEAR	Alle Elemente werden aus der LinkListCollection entfernt.	xxx
56	ILinkListCollection.COUNT	Gibt die Anzahl der aktuell in der LinkListCollection enthaltenen Datenelemente zurück.	xxx
57	ILinkListCollection.FIRST	Setzt den Index auf das erste Element und gibt den vorherigen Wert zurück.	xxx
58	ILinkListCollection.INDEX	Legt den Indexwert fest und gibt den vorherigen Wert zurück.	xxx
59	ILinkListCollection.INSERT	An der angegebenen Position wird ein Element zur LinkListCollection hinzugefügt.	xxx
60	ILinkListCollection.ITEM	Gibt das Element an der in der LinkListCollection angegebenen Position zurück.	xxx
61	ILinkListCollection.LAST	Setzt den Indexwert auf das letzte Element und gibt den vorherigen Wert zurück.	xxx
62	ILinkListCollection.NEXT	Gibt das nächste Element in der LinkListCollection zurück.	xxx
63	ILinkListCollection.PREVIOUS	Gibt das vorherige Element in der LinkListCollection zurück.	xxx
64	ILinkListCollection.REMOVE	Entfernt das Element an der angegebenen Position aus der LinkListCollection.	xxx

65	ILinkListCollection.REPLACE	Ersetzt das Element an der angegebenen Position durch ein neues Element in der LinkListCollection.	xxx
66	IPowerCollection.ADD	Ein Element und ein Schlüssel werden am Ende der PowerCollection hinzugefügt.	xxx
67	IPowerCollection.CLEAR	Entfernt alle Elemente und Schlüssel aus der PowerCollection.	xxx
68	IPowerCollection.CONTAINS	durchsucht die PowerCollection nach dem angegebenen Schlüssel.	xxx
69	IPowerCollection.COUNT	gibt die Anzahl der aktuell in der PowerCollection enthaltenen Datenelemente zurück	xxx
70	IPowerCollection.ENTRY	gibt das durch die Indexnummer angegebene PowerCollection-Element zurück.	xxx
71	IPowerCollection.FIRST	setzt den Index auf das erste Element und gibt den vorherigen Wert zurück.	xxx
72	IPowerCollection.INDEX	legt den Indexwert fest und gibt den vorherigen Wert zurück.	xxx
73	IPowerCollection.ITEM	gibt das mit dem angegebenen Schlüssel verknüpfte Element in der PowerCollection zurück.	xxx
80	IPowerCollection.LAST	setzt den Index auf das letzte Element und gibt den vorherigen Wert zurück.	xxx
81	IPowerCollection.NEXT	gibt das nächste Element in der PowerCollection zurück.	xxx
82	IPowerCollection.PREVIOUS	Gibt das vorherige Element in der PowerCollection zurück.	xxx
83	IPowerCollection.REMOVE	Entfernt das mit dem angegebenen Schlüssel verknüpfte Element aus der PowerCollection.	xxx
84	IPowerCollection.REPLACE	Ersetzt das mit dem angegebenen Schlüssel verknüpfte Element durch ein neues Element.	xxx
85	IPowerCollection.SORT	die Datenelemente in der PowerCollection werden basierend auf dem Text in den zugehörigen Schlüsseln sortiert.	xxx
86	IQueueCollection.CLEAR	alle Elemente werden aus der QueueCollection entfernt.	xxx
87	IQueueCollection.COUNT	gibt die Anzahl der aktuell in der QueueCollection enthaltenen Datenelemente zurück	xxx
88	IQueueCollection.DEQUEUE	das Element an der "ältesten" Position in der QueueCollection es wird zurückgegeben.	xxx
89	IQueueCollection.ENQUEUE	das angegebene Element wird der QueueCollection an der "neuesten" Position hinzugefügt.	xxx
90	IStackCollection.CLEAR	alle Elemente werden aus der StackCollection entfernt.	xxx
91	IStackCollection.COUNT	gibt die Anzahl der aktuell in der StackCollection enthaltenen Datenelemente zurück	xxx
92	IStackCollection.POP	das oberste Element des Stapels (das zuletzt hinzugefügte Element) wird zurückgegeben.	xxx

93	IStackCollection.PUSH	das angegebene Element wird der StackCollection an der obersten Position des Stacks hinzugefügt.	xxx
----	-----------------------	--	-----

3. COM Operationen

Zum Erstellen und Verwalten von COM-Clients können folgende Funktionen verwendet werden:

LfdNr	Befehl	Beschreibung	Seite
94	#COM DOC	gibt eine Hilfe-Zeichenfolge an, die normalerweise eine allgemeine Beschreibung des COM-Servers enthält	xxx
95	#COM HELP	gibt den Namen der zugehörigen Hilfedatei und den Hilfe-Kontextcode an	xxx
96	#COM NAME	gibt den Namen des Servers und die Versionsnummer an	xxx
97	#COM GUID	gibt die GUID an, die die gesamte Anwendung oder Bibliothek identifiziert (APPID oder LIBID)	xxx
98	ACODE\$	konvertiert eine Unicode-Zeichenfolge in eine ANSI-Zeichenfolge	xxx
99	CLASS/END CLASS	erstellt den Code und die Daten für ein Objekt.	xxx
100	CLSID\$	gibt eine 16 Byte (128 Bit) lange GUID-Zeichenfolge zurück, die eine CLSID enthält.	xxx
101	ENUM/END ENUM	erstellt eine Gruppe logisch verknüpfter numerischer Gleichungen	xxx
102	EVENT SOURCE	deklariert eine Ereignisschnittstelle innerhalb einer CLASS-Definition.	xxx
103	EVENTS	fügt einen Ereignishandler einer Ereignisquelle hinzu oder trennt ihn von dieser	xxx
104	FOR EACH/NEXT	definiert eine Schleife von Programmanweisungen, die jedes Mitglied einer PowerCollection oder LinkListCollection nacheinander prüfen und darauf reagieren kann	xxx
105	GUID\$	gibt eine 16 Byte (128 Bit) lange global eindeutige Kennung (GUID) zurück	xxx
106	GUIDTXT\$	gibt eine 38 Byte lange, für Menschen lesbare GUID/UUID-Zeichenfolge zurück	xxx
105	IDISPINFO	legt zusätzliche Informationen zu bestimmten Versandstatus-Codes für die Funktion OBJRESULT fest und gibt sie zurück	xxx
106	INSTANCE	deklariert INSTANCE-Variable, die für jedes Objekt eindeutig sind	xxx
107	INTERFACE / END INTERFACE	Block (Direct) deklariert eine direkte Objektschnittstelle und ihre Mitgliedsmethoden/-eigenschaften	xxx

108	INTERFACE / END INTERFACE block (IDBind)	deklariert eine Dispatch-Schnittstelle und deren Mitgliedsmethoden/-eigenschaften zum Zweck der ID-Bindung an eine Dispatch-COM-Schnittstelle	XXX
109	ISINTERFACE	ermittelt, ob ein Objekt eine bestimmte Schnittstelle unterstützt	XXX
110	ISNOTHING	ermittelt den aktuellen Status einer bestimmten Objektvariablen	XXX
111	ISOBJECT	ermittelt den aktuellen Status einer bestimmten Objektvariablen	XXX
112	LET (with Objects)	weist einer Objektvariablen eine Objektreferenz zu	XXX
113	LET (with Variants)	weist einer Variablen oder einem Variant einen Wert zu	XXX
114	ME	Pseudo-Objektvariable zum Verweisen auf das aktuelle Objekt	XXX
115	METHOD / END METHOD	definiert eine METHOD-Prozedur innerhalb einer CLASS	XXX
116	MYBASE	Pseudo-Objektvariable zum Verweisen auf das geerbte übergeordnete Objekt	XXX
117	OBJACTIVE	gibt True / False des Ausführungsstatus eines COM EXE-Objekts zurück	XXX
118	OBJECT GET	den Wert einer Dispatch- Schnittstellen-Member-Eigenschaft abrufen oder lesen	XXX
119	OBJECT LET	zuweisen oder schreiben eines Werts in eine Dispatch-Schnittstellen-Member-Eigenschaft.	XXX
120	OBJECT SET	einen Wert einer Dispatch-Schnittstellen-Member-Eigenschaft zuweisen oder einen Wert schreiben, der einen Verweis auf ein Objekt enthält	XXX
121	OBJECT CALL	eine Member-Methode einer Dispatch-Schnittstelle aufrufen oder sie ausführen	XXX
122	OBJECT RAISEEVENT	eine Member-Methode einer Ereignis-Dispatch-Schnittstelle aufrufen oder ausführen	XXX
123	OBJEQUAL	prüfen, ob Objektvariable auf dasselbe Objekt verweist	XXX
124	OBJPTR	gibt einen Objektzeiger einer angegebenen OBJECT-Variablen zurück	XXX
125	OBJRESULT	gibt das Ausführungsergebnis der letzten OBJECT-Anweisung zurück	XXX
126	OBJRESULT\$	gibt eine Zeichenfolge zurück, die einen OBJRESULT (hResult)-Code beschreibt	XXX
127	PROGID\$	gibt die alphanumerische PROGID-Zeichenfolge (Text) einer gegebenen CLSID zurück	XXX
128	PROPERTY GET	einen Datenwert aus dem Objekt abrufen	XXX
129	PROPERTY SET	einen Datenwert einem Objekt zuweisen	XXX
130	RAISEEVENT	Ereignishandler-Code aufrufen	XXX
131	RESET	einen Variant leeren (%VT_EMPTY)	XXX
132	UCODE\$	konvertiert eine ANSI-Zeichenfolge in eine Unicode-Zeichenfolge	XXX
133	VARIANT#	gibt den in einer Variant-Variablen enthaltenen numerischen Wert zurück	XXX

134	VARIANT\$	gibt den in einer Variant-Variablen enthaltenen dynamischen ANSI-String-Wert zurück	xxx
135	VARIANT\$\$	gibt den Unicode-String-Wert zurück, der in einer Variant-Variablen enthalten ist	xxx
136	VARIANTVT	den internen Datentyp der in einem Variant gespeicherten Daten bestimmen,	xxx

4. Kommunikations-Steuerung

Für die externe Kommunikation können folgende Funktionen genutzt werden:

LfdNr	Befehl	Beschreibung	Seite
137	COMM	den Wert oder des Status eines Kommunikations-Parameters abrufen	xxx
138	COMM CLOSE	einen offenen seriellen Port schließen	xxx
139	COMM LINE	eine CR/LF-terminierte "Zeile" von Daten von einem seriellen Port empfangen	xxx
140	COMM OPEN	einen seriellen Port öffnen	xxx
141	COMM PRINT	eine "Zeile" binärer Daten über einen seriellen Port senden	xxx
142	COMM RECV	von einem seriellen Port Binärdaten empfangen	xxx
143	COMM RESET	die Flusskontrolle für einen bestimmten seriellen Port deaktivieren	xxx
144	COMM SEND	eine Zeichenfolge binärer Daten über einen seriellen Port senden	xxx
145	COMM SET	Kommunikationsoptionen für einen seriellen Port festlegen	xxx
146	COMM TIMEOUT	die Zeit begrenzen, die zum Abschließen eines COMM-Vorgangs benötigt wird.	xxx
147	EOF	gibt den Datei-Ende-Status einer Datei, seriellen oder TCP/UDP-Übertragung zurück	xxx
148	FREEFILE	gibt die nächste verfügbare <i>PowerBASIC</i> -Dateinummer zurück	xxx
149	HOST ADDR	einen Hostnamen in eine entsprechende IP-Adresse übersetzen	xxx
150	HOST NAME	eine IP-Adresse in einen entsprechenden Hostnamen übersetzen	xxx
151	OPEN	eine Datei oder ein Geräts zum Lesen oder Schreiben vorbereiten	xxx
152	TCP ACCEPT	eine eingehende Anfrage für die TCP-Kommunikation akzeptieren	xxx
153	TCP CLOSE	einen zuvor geöffneten TCP/IP-Port schließen	xxx
154	TCP LINE INPUT	eine Textzeile von einem angegebenen TCP/IP-Port empfangen	xxx
155	TCP NOTIFY	festlegen, welche TCP/IP-Ereignisse Benachrichtigungs-Nachrichten generieren sollen	xxx
156	TCP OPEN	einer App die Kommunikation mit einem TCP/IP-Server oder -Client ermöglichen	xxx
157	TCP PRINT	eine Zeichenfolge in ein angegebenes TCP/IP schreiben	xxx

158	TCP_RECV	Daten von einem angegebenen TCP/IP-Port empfangen	xxx
159	TCP_SEND	eine Zeichenfolge an einen angegebenen TCP/IP-Port schreiben	xxx
160	UDP_CLOSE	einen zuvor geöffneten UDP-Socket schließen	xxx
161	UDP_NOTIFY	Festlegen, welche TCP/IP-Ereignisse Benachrichtigungs-Nachrichten generieren sollen	xxx
162	UDP_OPEN	einen Socket zur Kommunikation mit einem UDP-Server oder -Client erstellen	xxx
163	UDP_RECV	Daten von einem zuvor geöffneten UDP-Port empfangen	xxx
164	UDP_SEND	eine Datenzeichenfolge über einen zuvor geöffneten UDP-Socket senden	xxx

5. Compiler Operationen

Die folgenden Funktionen manipulieren die Arbeitsweise des Compilers:

LfdNr	Befehl	Beschreibung	Seite
165	#ALIGN	richtet die nächste Anweisung an einer Grenze aus.	xxx
166	%DEF	ermittelt, ob zuvor ein Equate definiert wurde	xxx
167	%PB_COMPILETIME	enthält Datum und Uhrzeit der Kompilierung	xxx
168	#BLOAT	künstliches Erhöhen der Disk-Image-Größe eines kompilierten Programms	xxx
169	#COMPILE	bestimmt, welcher Dateityp vom Compiler erstellt wird	xxx
170	#DEBUG_CODE	Compiler-Direktive zum Unterdrücken der Generierung von Debugcode	xxx
171	#DEBUG_DISPLAY	zeigt eine Meldung an, wenn ein nicht abgefangener Laufzeitfehler auftritt	xxx
172	#DIM	gibt an, ob Variablen vor der Verwendung deklariert werden müssen	xxx
173	#EXPORT	deklariert eine Subfunktion mit dem Attribut EXPORT	xxx
174	#IF	definiert Abschnitte des Quellcodes, die kompiliert oder ignoriert werden sollen	xxx
175	#LINK	verknüpft eine vorkompilierte Static Link Library (SLL) mit dem Hostprogramm	xxx
176	#MESSAGES	gibt an, welche Nachrichten an eine ControlCallback-Funktion gesendet werden sollen	xxx
177	#OPTIMIZE	wählt die Optimierung, die auf das Programm angewendet werden soll	xxx
178	#OPTION	verschiedene Compileroptionen festlegen	xxx
179	#REGISTER	steuert die automatische Zuweisung von Registervariablen	xxx
180	#STACK	legt die maximal mögliche Stack-Größe fest	xxx
181	#TOOLS	Aktivieren/Deaktivieren integrierter Entwicklungstools im kompilierten Code	xxx

182	#UNIQUE	gibt an, ob eindeutige Variable-Namen erforderlich sind	xxx
183	DECLARE	explizit eine Sub oder eine Function deklarieren	xxx
184	DEFBYT	den Standard-Variablentyp als Byte deklarieren	xxx
185	DEFCUR	den Standard-Variablentyp als Währung deklarieren	xxx
186	DEFCUX	den Standardvariablentyp als “Erweiterte Währung“ deklarieren	xxx
187	DEFDBL	den Standard-Variablentyp als “doppelte Genauigkeit“ deklarieren	xxx
188	DEFDWD	den Standard-Variablentyp als “Double-Word“ deklarieren	xxx
189	DEFEXT	den Standard-Variablentyp als erweiterte Genauigkeit deklarieren	xxx
190	DEFINT	den Standard-Variablentyp als Integer deklarieren	xxx
191	DEFLNG	den Standard-Variablentyp als Long-Integer deklarieren	xxx
192	DEFQUD	den Standard-Variablentyp als Quad-Integer deklarieren	xxx
193	DEFSNG	den Standard-Variablentyp als Single-Precision deklarieren	xxx
194	DEFSTR	den Standard-Variablentyp als String deklarieren	xxx
195	DEFWRD	den Standardvariablentyp als word deklarieren	xxx
196	DIM	deklarieren und dimensionieren von Arrays, Skalarvariablen und Zeigern	xxx
197	DLLMAIN	Funktion, die von Windows jedes Mal aufgerufen wird, wenn eine DLL in den Speicher geladen und aus dem Speicher entladen wird	xxx
198	ERASE	Array-Speicher freigeben	xxx
199	GLOBAL	deklarieren von globalen (gemeinsamen) Variablen zwischen Subs, Functions, CLASSES, Methoden und Eigenschaften	xxx
200	INSTANCE	deklarieren von Instanz-Variablen, die für jedes Objekt eindeutig sind	xxx
201	LIBMAIN	Funktion, die von Windows jedes Mal aufgerufen wird, wenn eine DLL in den Speicher geladen und aus dem Speicher entladen wird	xxx
202	LOCAL	lokale Variablen in einer Sub, einer Funktion, einer Methode oder einer Eigenschaft deklarieren	xxx
203	MACRO	einen ein- oder mehrzeiligen Textersatzblock definieren	xxx
204	OPTION EXPLICIT	die explizite Deklaration aller Variablen erzwingen	xxx
205	PBLIBMAIN	Funktion, die von Windows jedes Mal aufgerufen wird, wenn eine DLL in den Speicher geladen und aus dem Speicher entladen wird	xxx
206	PBMAIN	die anfängliche Einstiegspunkt-Funktion für eine Anwendung definieren	xxx
207	PREFIX/END PREFIX	führt eine Reihe von Anweisungen aus, die jeweils vordefinierten Quellcode verwenden	xxx
208	PROFILE	ein Ausführungszeitprofil der Subs, Funktionen, Methoden und Eigenschaften erfassen	xxx

209	REDIM	dynamische Arrays deklarieren, Speicher zuweisen, ihn neu zuweisen und die Zuordnung freigeben	xxx
210	REGISTER	lokale Registervariable innerhalb einer Sub, Funktion, Methode oder Eigenschaft definieren	xxx
211	STATIC	statische Variablen innerhalb einer Sub, einer Funktion, einer Methode oder einer Eigenschaft deklarieren	xxx
212	STRPTR	gibt die Adresse der Daten zurück, die in einer Zeichenfolge variabler Länge enthalten sind	xxx
213	VARPTR	gibt die 32-Bit-Adresse eines Variable- oder String-Handles zurück	xxx
214	WINMAIN	die anfängliche Einstiegspunktfunktion für eine Anwendung definieren	xxx

6. Debugging und Fehlerkontrolle

Die folgenden Funktionen können zum Abfangen und Verwalten von Fehlerzuständen verwendet werden:

LfdNr	Befehl	Beschreibung	Seite
215	#DEBUG CODE	Compiler-Direktive zum Unterdrücken der Generierung von Debug-Code	xxx
216	#DEBUG DISPLAY	zeigt eine Meldung an, wenn ein nicht abgefangener Laufzeitfehler auftritt	xxx
217	#DEBUG ERROR	Steuerung der Generierung von Fehlerprüfcode	xxx
218	#DEBUG PRINT	Informationen im Debug-Fenster der IDE anzeigen	xxx
219	#DIM	gibt an, ob Variablen vor der Verwendung deklariert werden müssen	xxx
220	#STACK	legt die maximal mögliche Stack-Größe fest	xxx
221	#TOOLS	aktivieren / deaktivieren der integrierten Entwicklungstools im kompilierten Code	xxx
222	CALLSTK	erfassen einer Darstellung der Stack-Frames im Aufruf-Stack	xxx
223	CALLSTK\$	abrufen der Details eines bestimmten Stack-Frames.	xxx
224	CALLSTKCOUNT	die Anzahl der Stack-Frames im Aufruf-Stack abrufen	xxx
225	ERL	gibt die Zeilennummer des aktuellsten Laufzeitfehlers zurück	xxx
226	ERL\$	gibt die letzte Bezeichnung, Zeilennummer oder den letzten Prozedurnamen zurück, der vor dem letzten Fehler ausgeführt wurde	xxx
227	ERR	gibt den Fehlercode des letzten Laufzeitfehlers zurück	xxx
228	ERRCLEAR	gibt den Fehlercode des letzten Laufzeitfehlers zurück und löscht ihn	xxx
229	ERROR	veranlasst die Generierung eines bestimmten Laufzeitfehlers und setzt ERR	xxx

230	ERROR\$	gibt eine Zeichenfolge zurück, die den beschreibenden Namen eines Fehlers enthält	xxx
231	FILENAME\$	gibt den Dateisystemnamen einer geöffneten Datei zurück	xxx
232	FUNCNAME\$	gibt den Namen der aktuellen Sub / Methode / Eigenschaft zurück	xxx
233	ON ERROR	eine Fehlerbehandlungsroutine angeben, aktivieren/deaktivieren von Trapping	xxx
234	OPTION EXPLICIT	erzwingen der expliziten Deklaration aller Variablen	xxx
235	PROFILE	erfassen eines Ausführungszeitprofils der Subs, Funktionen, Methoden und Eigenschaften	xxx
236	RESUME	fortsetzen der Ausführung nach Fehlerbehandlung mit ON ERROR GOTO	xxx
237	RESUME FLUSH	die Ausführung wird in der Zeile unmittelbar nach RESUME FLUSH fortgesetzt	xxx
238	RESUME NEXT	die Ausführung wird in der Zeile fortgesetzt, die unmittelbar auf die Zeile folgt, die den Fehler verursacht hat	xxx
239	RESUME <Label>	die Ausführung wird an der angegebenen Label-Position fortgesetzt	xxx
240	TRACE	erfassen des genauen Ausführungsfluss in einem Modul	xxx
241	TRY/END TRY	eine strukturierte Methode zum Auffangen und Reagieren auf Fehler	xxx

7. Befehle der dynamischen Dialogtools

Die folgenden Funktionen können zum Erstellen von GUI-Anwendungsschnittstellen verwendet werden:

LfdNr	Befehl	Beschreibung	Seite
242	ACCEL ATTACH	einem DDT-Dialog eine Tabelle mit Tastatur-Beschleunigern hinzufügen	xxx
243	CALLBACK FUNCTION	einen Dialog-/Steuerungs-Callback-Funktionsblock definieren	xxx
244	CB.CTL	gibt die numerische ID des Steuerelements zurück, das eine Callback-Nachricht sendet	xxx
245	CB.CTLMSG	gibt den numerischen Benachrichtigungsparameter zurück	xxx
246	CB.HNDL	gibt das Fenster-Handle des übergeordneten Dialogs zurück, der die Nachricht empfängt	xxx
247	CB.LPARAM	gibt den numerischen Wert des lParam-&-Parameters der Nachricht zurück	xxx
248	CB.MSG	gibt den numerischen Wert der vom Anrufer gesendeten Nachricht zurück	xxx

249	CB.WPARAM	gibt den numerischen Wert des wParam-Parameters der Nachricht zurück	xxx
250	CB.NMCODE	gibt den numerischen Wert der Benachrichtigung zurück, die das aufgetretene Ereignis beschreibt	xxx
251	CB.NMHDR	gibt die Adresse (einen Zeiger) zum NMHDR-UDT für diese Benachrichtigungs-Nachricht zurück	xxx
252	CB.NMHDR\$	gibt den Inhalt des NMHDR-UDT als dynamische Zeichenfolge zurück	xxx
253	CB.NMHWND	gibt das Handle des Steuerelements zurück, das diese Nachricht gesendet hat	xxx
254	CB.NMID	gibt die dem Steuerelement zugewiesene ID-Nummer zurück	xxx
255	CLIPBOARD GET BITMAP	eine Bitmap wird aus der Zwischenablage kopiert und in einer neu erstellten GRAFIK-BITMAP gespeichert	xxx
256	CLIPBOARD GET OEMTEXT	eine Textzeichenfolge wird aus der Zwischenablage abgerufen. Bei Bedarf wird sie in das OEM-Textformat konvertiert	xxx
257	CLIPBOARD GET TEXT	eine Textzeichenfolge wird aus der Zwischenablage abgerufen. Bei Bedarf wird sie in das ASCII-Textformat konvertiert	xxx
258	CLIPBOARD GET UNICODE	eine Textzeichenfolge wird aus der Zwischenablage abgerufen. Bei Bedarf wird sie in das Unicode-Textformat konvertiert	xxx
259	CLIPBOARD RESET	der Inhalt der Zwischenablage wird gelöscht	xxx
260	CLIPBOARD SET BITMAP	kopiert eine GRAFISCHE BITMAP in die Zwischenablage	xxx
261	CLIPBOARD SET OEMTEXT	kopiert eine OEM-Textzeichenfolge in die Zwischenablage	xxx
262	CLIPBOARD SET TEXT	kopiert eine ASCII-Textzeichenfolge in die Zwischenablage	xxx
263	CLIPBOARD SET UNICODE	Kopiert eine Unicode-Textzeichenfolge in die Zwischenablage	xxx
264	COMBOBOX ADD	fügt einer COMBOBOX einen Zeichenfolgen-Wert hinzu	xxx
265	COMBOBOX DELETE	entfernt eine Zeichenfolge aus einer COMBOBOX	xxx
266	COMBOBOX FIND	Zeichenfolgen in der COMBOBOX werden durchsucht, um die erste Zeichenfolge zu finden, die mit den angegebenen Zeichen beginnt	xxx
267	COMBOBOX FIND EXACT	Zeichenfolgen in der COMBOBOX werden durchsucht, um die erste Zeichenfolge zu finden, die genau mit den angegebenen Zeichen übereinstimmt	xxx
268	COMBOBOX GET COUNT	die Anzahl der Elemente im Listefeld der COMBOBOX wird abgerufen	xxx
269	COMBOBOX GET SELCOUNT	die Anzahl der ausgewählten Elemente im Listefeld der COMBOBOX wird abgerufen	xxx
270	COMBOBOX GET SELECT	der Index des aktuell ausgewählten Elements im Listefeld der COMBOBOX wird abgerufen	xxx
271	COMBOBOX GET STATE	ein Datenelement wird überprüft, um festzustellen, ob es aktuell ausgewählt ist	xxx

272	COMBOBOX GET TEXT	ruft den Standardtext aus einer COMBOBOX ab	xxx
273	COMBOBOX GET USER	ruft den Wert im Benutzer-Datenbereich der COMBOBOX ab	xxx
274	COMBOBOX INSERT	fügt an einer angegebenen Stelle ein neues Datenelement ein	xxx
275	COMBOBOX RESET	entfernt alle Zeichenfolgen aus einer COMBOBOX	xxx
276	COMBOBOX SELECT	eine Zeichenfolge in einer COMBOBOX auswählen und zur Standardauswahl.machen	xxx
277	COMBOBOX SET TEXT	ersetzt die Zeichenfolge für ein bestimmtes Datenelement durch eine neue Zeichenfolge	xxx
278	COMBOBOX SET USER	setzt einen Wert im Benutzer-Datenbereich der COMBOBOX	xxx
279	COMBOBOX UNSELECT	alle Elemente in einer COMBOBOX werden auf einen nicht ausgewählten Status gesetzt	xxx
280	CONTROL ADD	hinzufügen eines benutzerdefinierten Steuerelements zu einem DDT-Dialog	xxx
281	CONTROL ADD BUTTON	hinzufügen einer Befehlsschaltfläche zu einem Dialogfeld	xxx
282	CONTROL ADD CHECK3STATE	einem Dialogfeld ein automatisches 3-Status-Kontrollkästchen hinzufügen	xxx
283	CONTROL ADD CHECKBOX	ein Kontrollkästchen einem Dialogfeld.hinzufügen	xxx
284	CONTROL ADD COMBOBOX	ein Kombinationsfeld einem Dialogfeld hinzufügen	xxx
285	CONTROL ADD FRAME	ein Frame-Steuerelement einem Dialogfeld hinzufügen.	xxx
286	CONTROL ADD GRAPHIC	ein grafisches Steuerelement einem Dialogfeld hinzufügen.	xxx
287	CONTROL ADD HEADER	ein Header-Steuerelement einem Dialogfeld hinzufügen	xxx
288	CONTROL ADD IMAGE	ein Bildsteuerelement ohne Größenänderung einem Dialogfeld hinzufügen	xxx
289	CONTROL ADD IMAGEX	ein Bildsteuerelement einem Dialogfeld hinzufügen	xxx
290	CONTROL ADD IMGBUTTON	eine Bildschaltfläche ohne Größenänderung einem Dialogfeld hinzufügen	xxx
291	CONTROL ADD IMGBUTTONX	eine Bildschaltfläche einem Dialogfeld hinzufügen.	xxx
292	CONTROL ADD LABEL	eine Textbeschriftung einem Dialogfeld hinzufügen	xxx
293	CONTROL ADD LINE	ein Zeilensteuerelement (Linie) einem Dialogfeld hinzufügen.	xxx
294	CONTROL ADD LISTBOX	ein Listenfeld-Steuerelement einem Dialogfeld hinzufügen.	xxx
295	CONTROL ADD LISTVIEW	ein ListView-Steuerelements einem Dialogfeld hinzufügen	xxx
296	CONTROL ADD OPTION	eine Optionsschaltfläche einem Dialogfeld hinzufügen	xxx
297	CONTROL ADD PROGRESSBAR	ein ProgressBar-Steuerelement einem Dialogfeld. hinzufügen	xxx
298	CONTROL ADD SCROLLBAR	ein Bildlaufleisten-Steuerelement einem Dialogfeld hinzufügen.	xxx
299	CONTROL ADD STATUSBAR	ein StatusBar-Steuerelement einem Dialogfeld hinzufügen.	xxx
300	CONTROL ADD TAB	ein Registerkarten-Steuerelement einem Dialogfeld hinzufügen	xxx

301	CONTROL ADD TEXTBOX	ein Textfeld-Steuererelements zu einem Dialogfeld hinzufügen.	xxx
302	CONTROL ADD TOOLBAR	ein ToolBar-Steuererelement einem Dialogfeld hinzufügen.	xxx
303	CONTROL ADD TREEVIEW	ein TreeView-Steuererelement einem Dialogfeld hinzufügen.	xxx
304	CONTROL DISABLE	ein Steuererelement deaktivieren, sodass es keine Benutzerinteraktion mehr akzeptiert	xxx
305	CONTROL ENABLE	ein Steuererelement aktivieren, damit es Benutzerinteraktionen empfangen kann	xxx
306	CONTROL GET CHECK	den Prüfstatus eines 3-Zustands-, Kontrollkästchen- oder Optionsfelds abrufen	xxx
307	CONTROL GET CLIENT	die Abmessungen des Clientbereichs eines Steuererelements ermitteln	xxx
308	CONTROL GET LOC	die Position des angegebenen Steuererelements in einem Dialogfeld ermitteln.	xxx
309	CONTROL GET SIZE	die Größe eines Steuererelements im angegebenen Dialogfeld ermitteln	xxx
310	CONTROL GET TEXT	den Text aus einem Steuererelement holen	xxx
311	CONTROL GET USER	den Wert aus dem Benutzerdatenbereich eines DDT-Steuererelements abrufen	xxx
312	CONTROL HANDLE	gibt das Fenster-Handle für eine bestimmte Steuererelement-ID zurück	xxx
313	CONTROL HIDE	ein Steuererelement unsichtbar machen	xxx
314	CONTROL KILL	entfernen eines Steuererelements aus einem Dialogfeld	xxx
315	CONTROL NORMALIZE	ein Steuererelement sichtbar machen	xxx
316	CONTROL POST	eine Nachricht in die Nachrichten-Warteschlange eines Steuererelements einfügen (nicht blockierend)	xxx
317	CONTROL REDRAW	die Neuzeichnung eines Steuererelements vorbereiten	xxx
318	CONTROL SEND	eine Nachricht an ein Steuererelement senden und auf deren Verarbeitung warten	xxx
319	CONTROL SET CHECK	den Prüfstatus für ein 3-Zustands- oder Kontrollkästchen-Steuererelement festlegen	xxx
320	CONTROL SET CLIENT	die Größe eines Steuererelements auf eine bestimmte Clientbereichsgröße ändern	xxx
321	CONTROL SET COLOR	die Vordergrund- und Hintergrundfarbe eines Steuererelements festlegen	xxx
322	CONTROL SET FOCUS	den Tastaturfokus auf das angegebene Steuererelement setzen	xxx
323	CONTROL SET FONT	eine Schriftart auswählen, die für ein bestimmtes Windows-Steuererelement verwendet werden soll.	xxx
324	CONTROL SET IMAGE	die in einem IMAGE-Steuererelement angezeigten Symbole oder Bitmaps ändern	xxx
325	CONTROL SET IMAGEX	die in einem IMAGEX-Steuererelement angezeigte Symbole oder Bitmaps ändern	xxx
326	CONTROL SET IMGBUTTON	das in einem IMGBUTTON-Steuererelement angezeigte Symbol oder Bitmap ändern	xxx
327	CONTROL SET IMGBUTTONX	das in einem IMGBUTTONX-Steuererelement angezeigte Symbol oder Bitmap ändern.	xxx

328	CONTROL SET LOC	das Steuerelement an eine neue Position im Dialogfeld verschieben	XXX
329	CONTROL SET OPTION	den Prüfstatus für eine Optionssteuerung (RadioControl) festlegen	XXX
330	CONTROL SET SIZE	die Größe eines Steuerelements ändern	XXX
331	CONTROL SET TEXT	den Text in einem Steuerelement ändern	XXX
332	CONTROL SET USER	den Wert im Benutzerdatenbereich eines DDT-Steuerelements setzen	XXX
333	CONTROL SHOW STATE	den sichtbaren Status eines Steuerelements ändern	XXX
334	DESKTOP GET CLIENT	die Größe des Clientbereichs des Desktops in Pixeln abrufen	XXX
335	DESKTOP GET LOC	die Position der oberen linken Ecke des Clientbereichs des Desktops in Pixel abrufen	XXX
336	DESKTOP GET SIZE	gibt die Größe des angegebenen Dialogs zurück	XXX
337	DIALOG DISABLE	deaktiviert einen Dialog, sodass er nicht mehr auf Benutzerinteraktionen reagiert	XXX
338	DIALOG DOEVENTS	verarbeitet ausstehende Fenster- oder Dialognachrichten für nicht modale Dialoge	XXX
339	DIALOG ENABLE	aktiviert einen Dialog, sodass er auf Benutzerinteraktionen reagiert	XXX
340	DIALOG END	schließt und zerstört den angegebenen Dialog	XXX
341	DIALOG DEFAULT FONT	die Standardschriftart und -punktgröße von DDT angeben.	XXX
342	DIALOG GET CLIENT	gibt die Clientgröße des angegebenen Dialogs zurück.	XXX
343	DIALOG GET LOC	gibt den Speicherort des angegebenen Dialogs zurück.	XXX
344	DIALOG GET SIZE	gibt die Größe des angegebenen Dialogs zurück	XXX
345	DIALOG GET TEXT	den Text in einem Dialog oder einer Fensterüberschrift abrufen	XXX
346	DIALOG GET USER	einen Wert aus dem Benutzerdatenbereich eines DDT-Dialogs abrufen.	XXX
347	DIALOG HIDE	macht einen Dialog unsichtbar	XXX
348	DIALOG MAXIMIZE	maximiert einen Dialog	XXX
349	DIALOG MINIMIZE	minimiert einen Dialog	XXX
350	DIALOG NEW	einen neuen Dialog im Speicher erstellen, bereit zur Anzeige	XXX
351	DIALOG NONSTABLE	einen Dialog instabil (schließbar) machen	XXX
352	DIALOG NORMALIZE	einen Dialog sichtbar machen	XXX
353	DIALOG PIXELS	konvertiert Pixel (Geräteeinheiten) in Dialogeinheiten.	XXX
354	DIALOG POST	eine Nachricht in der Dialognachrichten-Warteschlange (nicht blockierend) einfügen	XXX
355	DIALOG REDRAW	erzwingen, dass ein Dialog, sowie alle untergeordneten Steuerelemente sofort neu gezeichnet werden	XXX
356	DIALOG SEND	eine Nachricht an einen Dialog senden und warten, bis sie verarbeitet wird	XXX
357	DIALOG SET CLIENT	die Größe eines Dialogs ändern auf eine bestimmte Clientbereichs-Größe	XXX

358	DIALOG SET COLOR	die Hintergrundfarbe eines Dialogs auf eine bestimmte RGB-Farbe festlegen	xxx
359	DIALOG SET ICON	das Dialogsymbol in der Überschrift, als auch das in der ALT+TAB-Aufgabenliste angezeigte Symbol ändern	xxx
360	DIALOG SET LOC	ändert die Position eines Dialogs	xxx
361	DIALOG SET SIZE	ändert die Größe eines Dialogs	xxx
362	DIALOG SET TEXT	den Text in einem Dialog oder einer Fensterüberschrift festlegen	xxx
363	DIALOG SET USER	einen Wert im Benutzer-Datenbereich eines DDT-Dialogs setzen	xxx
364	DIALOG SHOW MODAL	einen modalen Dialog anzeigen und ihn aktivieren	xxx
365	DIALOG SHOW MODELESS	einen nichtmodalen Dialog anzeigen und ihn aktivieren	
366	DIALOG SHOW STATE	den sichtbaren Status eines Dialogs ändern	xxx
367	DIALOG STABILIZE	einen Dialog stabilisieren (nicht schließbar machen)	xxx
368	DIALOG UNITS	die Dialogeinheiten in Pixel konvertieren	xxx
369	DISPLAY BROWSE	einen Ordnerauswahldialog anzeigen, um die Auswahl des Benutzers zurückzugeben	xxx
370	DISPLAY COLOR	einen Farbauswahldialog anzeigen, um die Auswahl des Benutzers zurückzugeben	xxx
371	DISPLAY FONT	einen Auswahldialog anzeigen, um die Benutzerauswahl zurückzugeben	xxx
372	DISPLAY OPENFILE	einen OpenFileDialog-Anwahldialog anzeigen, um die Benutzerauswahl zurückzugeben	xxx
373	DISPLAY SAVEFILE	einen SaveFile-Auswahldialog anzeigen, um die Benutzerauswahl zurückzugeben	xxx
374	FONT END	eine Schriftart zerstören, wenn sie nicht mehr benötigt wird	xxx
375	FONT NEW	eine neue Schriftart erstellen zur Verwendung mit GRAPHIC PRINT, XPRINT usw.	xxx
376	HEADER GET COUNT	die Anzahl der Elemente in einem Header-Steurelement abrufen.	xxx
377	HEADER GET ITEM	eine HD_Item-Struktur abrufen, die ein Element in einem Header-Steurelement beschreibt	xxx
378	HEADER SEND	eine Nachricht an ein Header-Steurelement senden	xxx
379	HEADER SET ITEM	die Attribute des angegebenen Elements in einem Header-Steurelement festlegen	xxx
380	IMAGELIST ADD BITMAP	ein Bitmap-Bild der IMAGELIST hinzufügen	xxx
381	IMAGELIST ADD ICON	Ein ICON zur IMAGELIST hinzufügen	xxx
382	IMAGELIST ADD MASKED	eine Bitmap der ICON IMAGELIST hinzufügen	xxx
383	IMAGELIST GET COUNT	die Anzahl der Bilder in der IMAGELIST wird abgerufen	xxx
384	IMAGELIST KILL	die angegebene IMAGELIST wird zerstört	xxx
385	IMAGELIST NEW BITMAP	eine neue Bitmap-IMAGELIST-Struktur wird erstellt	xxx
386	IMAGELIST NEW ICON	eine neue ICON-IMAGELIST-Struktur wird erstellt	xxx
387	IMAGELIST SET OVERLAY	ein Bild angeben, das als Overlay verwendet werden soll	xxx
388	INPUTBOX\$	ein Dialogfeld mit einer Eingabeaufforderung anzeigen	xxx

389	ISMISSING	ermitteln, ob vom aufrufenden Code ein optionaler Parameter übergeben wurde	xxx
390	ISWIN	ermitteln, ob derzeit ein Steuerelement/Dialog/Fenster vorhanden ist	xxx
391	LISTBOX ADD	einem LISTBOX-Steuerelement einen STRING-Wert hinzufügen	xxx
392	LISTBOX DELETE	eine Zeichenfolge aus einem LISTBOX-Steuerelement entfernen	xxx
393	LISTBOX FIND	die Zeichenfolgen in der LISTBOX werden durchsucht, um die erste Zeichenfolge zu finden, die mit den angegebenen Zeichen beginnt	xxx
394	LISTBOX FIND EXACT	die Zeichenfolgen in der LISTBOX werden durchsucht, um die erste Zeichenfolge zu finden, die genau mit den angegebenen Zeichen übereinstimmt	xxx
395	LISTBOX GET COUNT	die Anzahl der Elemente in der LISTBOX wird abgerufen	xxx
396	LISTBOX GET SELCOUNT	die Anzahl der ausgewählten Elemente in der LISTBOX wird abgerufen.	xxx
397	LISTBOX GET SELECT	die LISTBOX wird durchsucht, um das erste ausgewählte Element zu finden	xxx
398	LISTBOX GET STATE	ein Datenelement wird überprüft, um festzustellen, ob es aktuell ausgewählt ist	xxx
399	LISTBOX GET TEXT	den Standardtext aus einem LISTBOX-Steuerelement abrufen	xxx
400	LISTBOX GET USER	den Wert im Benutzerdatenbereich der LISTBOX abrufen	xxx
401	LISTBOX INSERT	an einer angegebenen Stelle ein neues Datenelement einfügen	xxx
402	LISTBOX RESET	alle Zeichenfolgen aus einem Listenfeld.entfernen	xxx
403	LISTBOX SELECT	eine Zeichenfolge in einem Listenfeld auswählen und sie als Standardauswahl festlegen	xxx
404	LISTBOX SET TEXT	die Zeichenfolge für ein bestimmtes Datenelement durch eine neue Zeichenfolge ersetzen	xxx
405	LISTBOX SET USER	einen Wert im Benutzerdatenbereich der LISTBOX setzen	xxx
406	LISTBOX UNSELECT	ein angegebenes Datenelement in der LISTBOX wird auf einen <u>nicht</u> ausgewählten Zustand gesetzt.	xxx
407	LISTVIEW DELETE COLUMN	aus der LISTVIEW, wird einer Spalte gelöscht, einschließlich des zugehörigen Kopftextes (sofern vorhanden)	xxx
408	LISTVIEW DELETE ITEM	das angegebene Datenelement wird aus der LISTVIEW gelöscht	xxx
409	LISTVIEW FIND	in der LISTVIEW werden Zeichenfolgen durchsucht, um die erste Zeichenfolge zu finden, die mit den angegebenen Zeichen beginnt	xxx
410	LISTVIEW FIND EXACT	in der LISTVIEW werden Zeichenfolgen durchsucht, um die erste Zeichenfolge zu finden, die <u>genau</u> mit den angegebenen Zeichen übereinstimmt	xxx
411	LISTVIEW FIT CONTENT	die Breite der angegebenen Spalte wird an die Breite der in dieser Spalte angezeigten Datenelemente angepasst	xxx

412	LISTVIEW FIT HEADER	die Breite der angegebenen Spalte wird an die Breite der in dieser Spalte angezeigten Datenelemente und des Kopftextes oben in dieser Spalte angepasst	XXX
413	LISTVIEW GET COLUMN	die Breite der angegebenen Spalte wird aus der LISTVIEW abgerufen	XXX
414	LISTVIEW GET COUNT	die Anzahl der Datenelemente wird in der LISTVIEW abgerufen	XXX
415	LISTVIEW GET HEADER	der Text der Spaltenüberschrift wird aus der LISTVIEW abgerufen	XXX
416	LISTVIEW GET HEADERID	das LISTVIEW-Handle und die Header-ID des Steuerelements abrufen	XXX
417	LISTVIEW GET MODE	der Anzeigemodus des angegebenen LISTVIEW-Steuerelements wird abgerufen	XXX
418	LISTVIEW GET SELCOUNT	die Anzahl der ausgewählten Elemente in der LISTVIEW wird abgerufen	XXX
419	LISTVIEW GET STATE	ein Datenelement wird getestet, um festzustellen, ob es aktuell ausgewählt ist	XXX
420	LISTVIEW GET STYLEXX	die aktuelle Einstellung des erweiterten Stils der LISTVIEW-Steuerelemente abrufen	XXX
421	LISTVIEW GET TEXT	ein Zeichenfolgen-Datenelement wird aus dem LISTVIEW-Steuerelement abgerufen	XXX
422	LISTVIEW GET USER	den Wert im Benutzerdatenbereich der LISTVIEW abrufen	XXX
423	LISTVIEW INSERT COLUMN	für den Berichtsmodus der LISTVIEW wird eine neue vertikale Spalte definiert	XXX
424	LISTVIEW INSERT ITEM	ein neues Datenelement wird dem LISTVIEW-Steuerelement hinzugefügt	XXX
425	LISTVIEW RESET	alle Datenelemente werden aus dem angegebenen LISTVIEW-Steuerelement gelöscht	XXX
426	LISTVIEW SELECT	das angegebene String-Datenelement wird als ausgewählter Text für die LISTVIEW ausgewählt	XXX
427	LISTVIEW SET COLUMN	die Breite einer LISTVIEW-Spalte ändern	XXX
428	LISTVIEW SET HEADER	ein neuer Spaltenüberschrifts.Text wird über der angegebenen Spalte im LISTVIEW-Steuerelement angezeigt	XXX
429	LISTVIEW SET IMAGE	das angegebene Bild wird neben dem angegebenen Element angezeigt.	XXX
430	LISTVIEW SET IMAGE2	das angegebene Bild wird als sekundäres "Statusbild" neben dem primären Bild angezeigt	XXX
431	LISTVIEW SET IMAGELIST	eine IMAGELIST an die LISTVIEW anfügen	XXX
432	LISTVIEW SET MODE	den Anzeigemodus des angegebenen LISTVIEW-Steuerelements ändern	XXX
433	LISTVIEW SET OVERLAY	das angegebene Over lay-Bild wird über dem angegebenen Bild angezeigt	XXX
434	LISTVIEW SET STYLE	die aktuellen Einstellungen des erweiterten Stils der LISTVIEW-Steuerelemente ändern	XXX
435	LISTVIEW SET TEXT	für das angegebene Datenelement wird der Text (sofern vorhanden) durch neuen Text ersetzt	XXX

436	LISTVIEW SET USER	einen Wert im Benutzerdatenbereich der LISTVIEW festlegen	XXX
437	LISTVIEW SORT	alle Elemente in einer LISTVIEW werden sortiert	XXX
438	LISTVIEW UNSELECT	das angegebene Datenelement wird auf einen <u>nicht</u> ausgewählten Status gesetzt.	XXX
439	LISTVIEW VISIBLE	um sicherzustellen, dass das Datenelement sichtbar ist, wird das angegebene Datenelement bei Bedarf gescrollt	XXX
440	PROGRESSBAR GET POS	die aktuelle Position des PROGRESSBAR wird abgerufen	XXX
441	PROGRESSBAR GET RANGE	der aktuelle Bereich des PROGRESSBAR wird abgerufen	XXX
442	PROGRESSBAR SET POS	die aktuelle Position des PROGRESSBAR festlegen	XXX
443	PROGRESSBAR SET RANGE	die minimalen und maximalen Bereiche der PROGRESSBAR festlegen	XXX
444	PROGRESSBAR SET STEP	den Standard-Inkrementwert angeben, der von PROGRESSBAR STEP verwendet werden soll	XXX
445	PROGRESSBAR STEP	die aktuelle Position der PROGRESSBAR wird um den Standardwert erhöht	XXX
446	MENU ADD POPUP	einem vorhandenen Menü ein Popup-Untermenü hinzufügen	XXX
447	MENU ADD STRING	eine Zeichenfolge oder ein Trennzeichen zu einem vorhandenen Menü hinzufügen	XXX
448	MENU ATTACH	einem bestimmten Dialog ein Menü hinzufügen	XXX
449	MENU CONTEXT	ein schwebendes Kontextmenü erstellen	XXX
450	MENU DELETE	ein Menüelement aus einem vorhandenen Menü löschen	XXX
451	MENU DRAW BAR	die Menüleiste für einen bestimmten Dialog neu zeichnen	XXX
452	MENU GET STATE	den Status eines angegebenen Menüelements zurückgeben	XXX
453	MENU GET TEXT	den mit einem bestimmten Menüelement verknüpften Text zurückgeben	XXX
454	MENU NEW BAR	eine neue Menüleiste erstellen	XXX
455	MENU NEW POPUP	ein neues Popup-Menü erstellen	XXX
456	MENU SET STATE	den Status eines angegebenen Menüelements festlegen	XXX
457	MENU SET TEXT	den Text eines bestimmten Menüelements festlegen	XXX
458	MOUSEPTR	den Mauszeiger (Cursor) in eine neue Form ändern	XXX
459	SCROLLBAR GET PAGESIZE	die aktuelle Seitengröße abrufen	XXX
460	SCROLLBAR GET POS	die aktuelle Position der SCROLLBAR zurückgeben	XXX
461	SCROLLBAR GET RANGE	den aktuellen Bereich der SCROLLBAR zurückgeben	XXX
462	SCROLLBAR GET TRACKPOS	die aktuelle Position der SCROLLBAR abrufen	XXX
463	SCROLLBAR SET PAGESIZE	legt die aktuelle Seitengröße fest	XXX
464	SCROLLBAR SET POS	die aktuelle Position der Bildlaufleiste festlegen	XXX
465	SCROLLBAR SET RANGE	den Bereich der Bildlaufleiste festlegen	XXX
466	STATUSBAR SET PARTS	die Anzahl der Teile festlegen, die in der STATUSBAR angezeigt werden sollen	XXX
467	STATUSBAR SET TEXT	den Text zuweisen, der im angegebenen Bereich der STATUSBAR angezeigt werden soll	XXX

468	TAB DELETE	eine Seite aus dem TAB-Steuerelement löschen	xxx
469	TAB GET COUNT	die Anzahl der Seiten in einem TAB-Steuerelement zurückgeben	xxx
470	TAB GET DIALOG	das Handle des Dialogs für eine bestimmte Seite in einem TAB-Steuerelement abrufen	
471	TAB GET IMAGE	den Index des auf der angegebenen TAB-Seite angezeigten Bildes abrufen.	xxx
472	TAB GET PAGE	die Seitennummer des angegebenen TAB-Seitendialog-Handles abrufen	xxx
473	TAB GET SELECT	die aktuell ausgewählte Seite in einem TAB-Steuerelement zurückgeben	xxx
474	TAB GET TEXT	der auf der angegebenen Seitenregisterkarte angezeigte Text wird abgerufen.	xxx
475	TAB INSERT PAGE	einem TAB-Steuerelement eine neue Seite hinzufügen	xxx
476	TAB RESET	alle Seiten in einem TAB-Steuerelement löschen	xxx
477	TAB SELECT	in einem TAB-Steuerelement eine bestimmte Seite als aktive Seite auswählen	xxx
478	TAB SET IMAGE	das angegebene Bild wird auf der angegebenen Seitenregisterkarte angezeigt	xxx
479	TAB SET IMAGELIST	eine IMAGELIST zuweisen, die in einem TAB-Steuerelement verwendet werden soll	xxx
480	TAB SET TEXT	den angegebenen Text auf der Registerkarte der Seite anzeigen	xxx
481	TOOLBAR ADD BUTTON	einem TOOLBAR-Steuerelement eine Schaltfläche hinzufügen	xxx
482	TOOLBAR ADD SEPARATOR	einem TOOLBAR-Steuerelement ein Trennzeichen hinzufügen	xxx
483	TOOLBAR DELETE BUTTON	eine Schaltfläche aus einem TOOLBAR-Steuerelement löschen	xxx
484	TOOLBAR GET STATE	den Status einer Schaltfläche auf einem TOOLBAR-Steuerelement abrufen	xxx
485	TOOLBAR GET COUNT	die Anzahl der Schaltflächen auf einem TOOLBAR-Steuerelement abrufen	xxx
486	TOOLBAR SET IMAGELIST	eine IMAGELISTE an einem TOOLBAR-Steuerelement anfügen	xxx
487	TOOLBAR SET STATE	den Status einer Schaltfläche auf einem TOOLBAR-Steuerelement festlegen	xxx
488	TREEVIEW DELETE	ein Datenelement aus einem TREEVIEW-Steuerelement löschen	xxx
489	TREEVIEW GET BOLD	das Attribut "Fettdruck" für ein Datenelement wird abgerufen.	xxx
490	TREEVIEW GET CHECK	das Attribut "Häkchen" für ein Datenelement wird abgerufen.	xxx
491	TREEVIEW GET CHILD	gibt das Handle des ersten untergeordneten Elements eines angegebenen Datenelements zurück	xxx
492	TREEVIEW GET COUNT	die Anzahl der Datenelemente in einem TREEVIEW wird abgerufen	xxx
493	TREEVIEW GET EXPANDED	das erweiterte Attribut für das Datenelement wird abgerufen	xxx

494	TREEVIEW GET NEXT	gibt das Handle des nächsten “Geschwister“-Datenelements zurück	XXX
495	TREEVIEW GET PARENT	das Handle des übergeordneten Elements für ein angegebenes Datenelement wird zurückgegeben	XXX
496	TREEVIEW GET PREVIOUS	gibt das Handle des vorherigen “Geschwister“-Datenelements zurück	XXX
497	TREEVIEW GET ROOT	das Handle des allerersten Datenelements (ganz oben) im TREEVIEW wird abgerufen	XXX
498	TREEVIEW GET SELECT	das Handle des aktuell ausgewählten Datenelements wird abgerufen	XXX
499	TREEVIEW GET TEXT	der Text eines bestimmten Datenelements wird abgerufen	XXX
500	TREEVIEW GET USER	den Wert im Benutzer-Datenbereich für ein bestimmtes Datenelement in der TREEVIEW abrufen	XXX
501	TREEVIEW INSERT ITEM	einem TREEVIEW-Steuerelement ein neues Datenelement hinzufügen	XXX
502	TREEVIEW RESET	alle Datenelemente werden aus dem angegebenen TREEVIEW-Steuerelement gelöscht	XXX
503	TREEVIEW SELECT	ein bestimmtes Datenelement im TREEVIEW-Steuerelement auswählen	XXX
504	TREEVIEW SET BOLD	das Attribut “Fettdruck“ für ein bestimmtes Datenelement festlegen	XXX
505	TREEVIEW SET CHECK	das Attribut “Häkchen“ für ein bestimmtes Datenelement festlegen	XXX
506	TREEVIEW SET EXPANDED	das erweiterte Attribut für ein bestimmtes Datenelement festlegen	XXX
507	TREEVIEW SET IMAGELIST	eine IMAGELIST an ein TREEVIEW-Steuerelement anfügen	XXX
508	TREEVIEW SET TEXT	für das angegebene Datenelement wird der Text (sofern vorhanden) durch neuen Text ersetzt	XXX
509	TREEVIEW SET USER	den Wert im Benutzer-Datenbereich für ein bestimmtes Datenelement im TREEVIEW-Steuerelement festlegen	XXX
510	TREEVIEW UNSELECT	alle Elemente im TREEVIEW-Steuerelement werden auf einen <u>nicht</u> ausgewählten Status gesetzt	XXX
511	WINDOW GET HANDLE	das Handle eines Fensters abrufen	XXX
512	WINDOW GET ID	die integrale ID des Fensters wird abgerufen	XXX
513	WINDOW GET PARENT	das Handle des übergeordneten Elements wird abgerufen	XXX
514	WINDOW GET STYLE	der Stil des Fensters wird abgerufen	XXX
515	WINDOW GET STYLEX	den erweiterten Stil des Fensters abrufen	XXX
516	WINDOW GET USER	den 32-Bit-Benutzerdatenwert abrufen, der mit dem Fenster verknüpft ist	XXX
517	WINDOW SET ID	die integrale ID des Fensters ändern	XXX
518	WINDOW SET STYLE	den Stil des Fensters ändern	XXX
519	WINDOW SET STYLEX	den erweiterten Stil des Fensters ändern	XXX
520	WINDOW SET USER	den mit dem Fenster verknüpften 32-Bit-Benutzer-Datenwert ändern	XXX

8. Dateibefehle

Die folgenden Funktionen können zur Manipulation von Dateien, Standard-E/A und Festplattendiensten verwendet werden:

LfdNr	Befehl	Beschreibung	Seite
521	CHDIR	das aktuelle (Standard-)Verzeichnis auf einem bestimmten Laufwerk ändern	xxx
522	CHDRIVE	das aktuelle Standardlaufwerk ändern	xxx
523	CLOSE	E/A (Eingabe/Ausgabe) zu/von einer Datei oder einem Gerät abschliessen	xxx
524	CURDIR\$	gibt das aktuelle Verzeichnis für ein bestimmtes Laufwerk zurück	xxx
525	DIR\$	einen Dateinamen zurückgeben, der mit der angegebenen Maske übereinstimmt	xxx
526	DIR\$ CLOSE	die Freigabe des FindNext-Handles des Betriebssystems erzwingen	xxx
527	DISKFREE	die Grösse des verfügbaren Speicherplatzes einer Festplatte in Bytes zurückgeben	xxx
528	DISKSIZE	den Gesamtspeicherplatz auf einer Festplatte in Bytes zurückgeben	xxx
529	EOF	den Dateiende-Status einer Datei, seriellen oder TCP/UDP-Übertragung zurückgeben	xxx
530	EXE	gibt den Pfad und/oder Namen des ausführenden Programms zurück	xxx
531	FIELD	eine FieldString-Variable an einen bestimmten Unterabschnitt eines zufälligen Dateipuffers oder einer DynamicString-Variablen binden	xxx
532	FIELD RESET	die FIELD-Zeichenfolge auf eine dynamische NUL-Zeichenfolge (Länge Null) zurücksetzen.	xxx
533	FIELD STRING	die Zeichenfolge "FIELD" in eine dynamische Zeichenfolge ändern, ihr jedoch zunächst die aktuellen Unterabschnittsdaten zuweisen	xxx
534	FILEATTR	Informationen zu einer geöffneten Datei zurückgeben	xxx
535	FILECOPY	eine Datei kopieren	xxx
536	FILENAME\$	den Datei-Systemnamen einer geöffneten Datei zurückgeben	xxx
537	FILESCAN	eine INPUT- oder BINARY-Datei schnell scannen, um Informationen zur Zeichenfolgenreihe zu erhalten.	xxx
538	FLUSH	den Dateipuffer auf die Festplatte leeren, um sicherzustellen, dass die Festplatteninformationen aktuell sind	xxx
539	FREEFILE	gibt die nächste verfügbare PowerBASIC-Dateinummer zurück	xxx
540	GET	einen Datensatz aus einer Direktzugriffs-Datei lesen	xxx
541	GET\$	einen ANSI-String aus einer im Binärmodus geöffneten Datei lesen	xxx

542	GET\$\$	WIDE-String-Daten aus einer im Binärmodus geöffneten Datei lesen	xxx
543	GETATTR	gibt die Dateisystem-Attribute einer Datenträger-Datei oder eines Verzeichnisses zurück	xxx
544	INPUT#	Variablen mit Daten aus einer sequentiellen Datei laden	xxx
545	ISFILE	feststellen, ob eine Datei vorhanden ist oder nicht	xxx
546	ISFOLDER	feststellen, ob ein Ordner vorhanden ist oder nicht.	xxx
547	KILL	eine Datei auf einem Datenträger löschen	xxx
548	LINE INPUT#	Zeile(n) aus einer sequenziellen Datei in eine Zeichenfolgen-Variable oder ein Array einlesen	xxx
549	LOC	die aktuellen Suchposition in einer geöffneten Datei auf der Festplatte ermitteln	xxx
550	LOCK	einen Teil oder die gesamte geöffnete Datei für exklusiven Zugriff sperren	xxx
551	LOF	gibt die Grösse einer geöffneten Datenträger-Datei zurück	xxx
552	MKDIR	ein Unterverzeichnis/einen Ordner (wie der DOS-Befehl MKDIR) erstellen	xxx
553	NAME	eine Datei oder ein Verzeichnis umbenennen (wie der DOS-Befehl REN)	xxx
554	OPEN	eine Datei oder ein Geräts zum Lesen oder Schreiben vorbereiten	xxx
555	PATHNAME\$	einen Pfad-/Dateinamen analysieren, um Komponententeile zu extrahieren.	xxx
556	PATHSCAN\$	eine Datei auf der Festplatte suchen und dabei den Pfad und/oder die Dateinamenteile zurückgeben	xxx
557	PRINT#	Daten auf ein Gerät oder in eine sequentielle Datei schreiben	xxx
558	PROFILE	eine Datei erstellen mit dem Zeitprofil von Subs, Functions, METHODS und PROBERTYS	xxx
559	PUT	einen Datensatz in eine Direktzugriffsdatei schreiben oder eine Variable in eine Binärdatei.	xxx
560	PUT\$	einen ANSI-String in eine im Binärmodus geöffnete Datei schreiben	xxx
561	PUT\$\$	eine WIDE-Unicode-Zeichenfolge in eine im Binärmodus geöffnete Datei schreiben	xxx
562	RMDIR	ein Datenträgerverzeichnis löschen (wie mit dem DOS-Befehl RMDIR)	xxx
563	SEEK	den Dateispeicherort bestimmen, an dem der nächste E/A-Vorgang stattfinden wird	xxx
564	SEEK	festlegen der Position in einer Datei für den nächsten Eingabe- oder Ausgabevorgang.	xxx
565	SETATTR	die Dateisystemattribute einer Datenträgerdatei oder eines Verzeichnisses festlegen	xxx
566	SETEOF	Kürzen/Erweitern einer Datei bis zu ihrer aktuellen Dateizeigerposition	xxx
567	SHELL	ein ausführbares Programm synchron/asynchron ausführen	xxx
568	UNLOCK	exklusive Zugriffssperren für eine Datei entfernen	xxx

569	WRITE#	Daten in eine sequentielle Datei in einem begrenzten Format ausgeben	xxx
-----	--------	--	-----

9. Ablaufsteuerung

Die folgenden Funktionen können zur Steuerung der Programmausführung/des Programmablaufs verwendet werden:

LfdNr	Befehl	Beschreibung	Seite
570	%DEF	feststellen, ob zuvor ein Equate definiert wurde	xxx
571	#IF	Abschnitte des Quellcodes definieren, die kompiliert oder ignoriert werden sollen	xxx
572	#TOOLS	die integrierten Entwicklungstools im kompilierten Code aktivieren / deaktivieren	xxx
573	CALL	eine Prozedur aufrufen (Sub, Function, Method, Property oder FastProc)	xxx
574	CALL DWORD	eine Prozedur indirekt aufrufen (Sub, Function, Method, Property oder FastProc)	xxx
575	CALLSTK	eine Darstellung der Stack-Frames im Aufrufstapel erfassen	xxx
576	CALLSTK\$	Details eines bestimmten Stack-Frames abrufen.	xxx
577	CALLSTKCOUNT	die Anzahl der Stack-Frames im Aufrufstapel abrufen	xxx
578	CHOOSE	gibt einen von mehreren Werten zurück, basierend auf dem Wert eines Index	xxx
579	CODEPTR	eine 32-Bit-Adresse eines Labels oder einer Prozedur erhalten	xxx
580	DLLMAIN	benutzerdefinierte Funktion, die aufgerufen wird, wenn eine DLL geladen/entladen wird	xxx
581	DO/LOOP	eine Gruppe von Anweisungen definieren, die wiederholt ausgeführt werden	xxx
582	END	beenden das Programm sofort	xxx
583	EXIT	die Programmausführung aus einer Blockstruktur.übertragen	xxx
584	FASTPROC/END FASTPROC	einen FastProc-Codeabschnitt definieren	xxx
585	FOR/NEXT	eine Schleife von Programmanweisungen definieren, die von einem Zähler gesteuert wird	xxx
586	FOR EACH/NEXT	eine Schleife von Programmanweisungen definieren, die jedes Mitglied einer PowerCollection oder LinkListCollection nacheinander untersuchen und darauf reagieren können	xxx
587	FUNCNAME\$	gibt den Namen des aktuellen Subs, der aktuellen Function, Method oder Property zurück.	xxx
588	FUNCTION/END FUNCTION	einen Funktionsblock definieren	xxx
589	GOSUB	eine lokale Subroutine aufrufen	xxx
590	GOSUB DWORD	eine lokale Subroutine indirekt aufrufen	xxx
591	GOTO	die Programmausführung an die durch ein Label identifizierte Anweisung übertragen	xxx
592	GOTO DWORD	die Ausführung indirekt auf ein lokales Label oder eine Zeilennummer übertragen	xxx

593	IF	eine Bedingung testen und danach eine oder mehrere Programmanweisungen ausführen	xxx
594	IF/END IF	einen IF/THEN/ELSE-Block mit mehreren Zeilen und Bedingungen erstellen	xxx
595	IIF	gibt einen von zwei Werten basierend auf einer True/False-Auswertung zurück	xxx
596	ISFALSE	gibt die logische Falschheit eines gegebenen Ausdrucks zurück	xxx
597	ISMISSING	ermittelt, ob vom aufrufenden Code ein optionaler Parameter übergeben wurde	xxx
598	ISNOTHING	ermittelt den aktuellen Status einer bestimmten Objektvariablen	xxx
599	ISOBJECT	bestimmt den aktuellen Status einer bestimmten Objektvariablen	xxx
600	ISTRUE	gibt die logische Wahrheit eines gegebenen Ausdrucks zurück	xxx
601	ITERATE	eine sofortige Iteration einer Schleifenstruktur starten	xxx
602	LIBMAIN	eine benutzerdefinierte Funktion wird aufgerufen, wenn eine DLL geladen/entladen wird.	xxx
603	MACRO	definiert einen ein- oder mehrzeiligen Textersetzungs-Block.	xxx
604	METHOD/END METHOD	definiert eine METHOD-Prozedur innerhalb einer CLASS.	xxx
605	ON ERROR	gibt eine Fehlerbehandlungsroutine an und aktiviert/deaktiviert das Trapping (Abfangen).	xxx
606	ON GOSUB	eine von mehreren Unterrouinen basierend auf einem numerischen Ausdruck aufrufen	xxx
607	ON GOTO	den Programmfluss senden, basierend auf einem Wert an eines von mehreren Labels	xxx
608	PBLIBMAIN	benutzerdefinierte Funktion aufrufen, wenn eine DLL geladen/entladen wird.	xxx
609	PBMAIN	die anfängliche Funktion für den Einstiegspunkt einer Anwendung definieren	xxx
610	PREFIX/END PREFIX	eine Reihe von Anweisungen ausführen, die jeweils den vordefinierten Quellcode verwenden	xxx
611	PROFILE	ein Ausführungszeitprofil der Subs, Functions, Methods und Property's erfassen.	xxx
612	PROPERTY/END PROPERTY	eine PROPERTY-Prozedur innerhalb einer CLASS definieren	xxx
613	RESUME	die Ausführung nach Fehlerbehandlung mit ON ERROR GOTO fortsetzen	xxx
614	RESUME FLUSH	die Ausführung wird in der Zeile unmittelbar nach RESUME FLUSH fortgesetzt	xxx
615	RESUME NEXT	die Ausführung wird in der Zeile fortgesetzt, die unmittelbar auf die Zeile folgt, die den Fehler verursacht hat	xxx
616	RESUME <Label>	die Ausführung wird an der angegebenen Label-Position fortgesetzt	xxx

617	RETURN	Rückkehr von einer (GOSUB)-Subroutine zu ihrem Aufrufer	xxx
618	RETURN FLUSH	die aktuellste Rücksprungadresse aus dem Systemstapel entfernen	xxx
619	SELECT CASE	den Programmfluss steuern, basierend auf dem Wert eines Ausdrucks	xxx
620	SLEEP	den aktuellen Thread für eine bestimmte Anzahl von Millisekunden pausieren	xxx
621	SUB/END SUB	definiert einen Sub-Block (Prozedurblock).	xxx
622	TRY/END TRY	eine strukturierte Methode zum Auffangen und Reagieren auf Fehler	xxx
623	WHILE/WEND	definiert einen Block von Anweisungen, die wiederholt ausgeführt werden	xxx
624	WINMAIN	definiert die anfängliche Funktion für den Einstiegspunkt einer Anwendung	xxx

10. Grafikkommandos

Zur Anzeige von Grafiken können folgende Befehle verwendet werden:

LfdNr	Befehl	Beschreibung	Seite
625	CONTROL ADD IMAGE	einem Dialog ein Bildsteuerelement (ohne Größenänderung) hinzufügen	xxx
626	CONTROL ADD IMAGEX	ein gestrecktes Bildsteuerelements zu einem Dialogfeld hinzufügen	xxx
627	CONTROL ADD IMGBUTTON	eine Bildschaltfläche zu einem Dialogfeld hinzufügen	xxx
628	CONTROL ADD IMGBUTTONX	einem Dialogfeld eine Schaltfläche zum Vergrößern eines Bilds hinzufügen	xxx
629	CONTROL ADD GRAPHIC	einem Dialog ein grafisches Steuerelement hinzufügen	xxx
630	BGR	konvertiert einen RGB-Farbwert in das BGR-Format	xxx
631	GRAPHIC(CANVAS.X)	die beschreibbare Breite des angehängten Grafikziels abrufen	xxx
632	GRAPHIC(CANVAS.Y)	die beschreibbare Höhe des angehängten Grafikziels abrufen	xxx
633	GRAPHIC(Cell.Size.X)	die Zeichenzellenbreite einschließlich des äußeren Zeilenabstands abrufen	xxx
634	GRAPHIC(Cell.Size.Y)	die Zeichenzellenhöhe einschließlich des äußeren Zeilenabstands abrufen	xxx
635	GRAPHIC(Chr.Size.X)	die Zeichenbreite auf dem Grafikziel abrufen	xxx
636	GRAPHIC(Chr.Size.Y)	die Zeichenhöhe auf dem Grafikziel abrufen	xxx
637	GRAPHIC(Client.X)	die Client-Breite des angehängten Grafikziels abrufen	xxx
638	GRAPHIC(Client.Y)	die Client-Höhe des angehängten Grafikziels abrufen	xxx
639	GRAPHIC(Clip.X)	die Breite des Clip-Bereichs abrufen	xxx
640	GRAPHIC(Clip.Y)	die Höhe des Clip-Bereichs ermitteln	xxx

641	GRAPHIC(COL)	die nächste Spaltendruckposition basierend auf der Zeilen- und Spaltenposition einer Textzelle abrufen	xxx
642	GRAPHIC(DC)	das Handle des DC (Gerätekontext) für das ausgewählte Grafikziel abrufen	xxx
643	GRAPHIC(INSTAT)	stellt fest, ob ein Tastaturzeichen bereit ist	xxx
644	GRAPHIC(LINES)	die Anzahl der Textzeilen abrufen, die auf das Grafikziel passen	xxx
645	GRAPHIC(LOC.X)	die horizontale Position des Grafikziels auf dem Desktop abrufen	xxx
646	GRAPHIC(LOC.Y)	die vertikale Position des Grafikziels auf dem Desktop abrufen	xxx
647	GRAPHIC(MIX)	den Farbmischmodus für das ausgewählte Grafikziel abrufen	xxx
648	GRAPHIC(OVERLAP)	den Status des Grafiküberlappungs-Modus abrufen	xxx
649	GRAPHIC(PIXEL . . .)	die Farbe des Pixels am angegebenen Punkt abrufen	xxx
650	GRAPHIC(POS.X)	die horizontale POS (letzter referenzierter Punkt) durch eine GRAPHIC-Anweisung abrufen	xxx
651	GRAPHIC(POS.Y)	die vertikale POS (letzter referenzierter Punkt) durch eine GRAPHIC-Anweisung abrufen	xxx
652	GRAPHIC(PPI.X)	die horizontale Auflösung des Anzeigegeräts in Punkten pro Zoll abrufen	xxx
653	GRAPHIC(PPI.Y)	die vertikale Auflösung des Anzeigegeräts in Punkten pro Zoll abrufen	xxx
654	GRAPHIC(ROW)	die nächste Zeilendruckposition abrufen, basierend auf der Zeilen- und Spaltenposition einer Textzelle	xxx
655	GRAPHIC(SCROLLTEXT)	den Status des GRAPHIC-ScrollText-Modus abrufen	xxx
656	GRAPHIC(SIZE.X)	die Gesamtbreite des ausgewählten GRAPHIC-Ziels abrufen	xxx
657	GRAPHIC(SIZE.Y)	die Gesamthöhe des ausgewählten GRAPHIC-Ziels abrufen	xxx
658	GRAPHIC(STRETCHMODE)	den Standard-Bitmap-Streckmodus für die angehängte DC abrufen	xxx
659	GRAPHIC(TEXT.SIZE.X . . .)	berechnet die Breite des zu druckenden Textes	xxx
660	GRAPHIC(TEXT.SIZE.Y . . .)	berechnet die Höhe des zu druckenden Textes	xxx
661	GRAPHIC(View.X)	die horizontale Position des virtuellen GRAPHIC-Ansichtsfensters abrufen	xxx
662	GRAPHIC(View.Y)	die vertikale Position des virtuellen GRAPHIC-Ansichtsfensters abrufen	xxx
663	GRAPHIC(WORDWRAP)	den Status des grafischen WordWrap-Modus abrufen	xxx
664	GRAPHIC(WRAP)	den Status des GRAPHIC-Umbruchmodus abrufen	xxx
665	GRAPHIC\$(CAPTION)	die Beschriftung aus einem Grafikfenster abrufen	xxx
666	GRAPHIC\$(INKEY\$)	ein Tastaturzeichen einlesen, wenn eines bereit ist	xxx
667	GRAPHIC\$(WAITKEY\$)	ein Tastaturzeichen einlesen oder eine erweiterte Taste und warten, bis eines bereit ist	xxx
668	GRAPHIC\$(WAITKEY\$. . .)	einen begrenzten Satz von Tastaturzeichen einlesen oder erweiterte Tasten mit einem optionalen Timeout-Wert	xxx

669	GRAPHIC ARC	einen Bogen zeichnen im ausgewählten Grafikfenster	xxx
670	GRAPHIC ATTACH	das Grafikziel (Fenster, Steuerelement oder Bitmap) auswählen, auf dem zukünftige Zeichenvorgänge stattfinden sollen	xxx
671	GRAPHIC BITMAP END	die ausgewählte Grafik-Bitmap schließen	xxx
672	GRAPHIC BITMAP LOAD	eine Speicher-Bitmap erstellen und ein Bild hinein laden	xxx
673	GRAPHIC BITMAP NEW	eine neue Speicher-Bitmap erstellen	xxx
674	GRAPHIC BOX	im ausgewählten Grafikfenster ein Kästchen zeichnen mit eckigen oder abgerundeten Ecken	xxx
675	GRAPHIC CELL	die nächste Druckposition einer Textzelle festlegen oder ruft sie abrufen	xxx
676	GRAPHIC CELL SIZE	die Zeichenzellengröße einschließlich des externen Zeilenabstands abrufen	xxx
677	GRAPHIC CHR SIZE	die Zeichengröße für die aktuelle Schriftart im ausgewählten Grafikfenster abrufen	xxx
678	GRAPHIC CLEAR	das gesamte ausgewählte Grafikfenster löschen, optional mit einer bestimmten Farbe und einem bestimmten Füllstil	xxx
679	GRAPHIC COLOR	die Vordergrundfarbe festlegen und optional die Hintergrundfarbe für verschiedene grafische Aussagen	xxx
680	GRAPHIC COPY	eine Bitmap in das ausgewählte Grafikziel kopieren	xxx
681	GRAPHIC DETACH	ein Grafikziel ablösen	xxx
682	GRAPHIC ELLIPSE	eine Ellipse oder einen Kreis im ausgewählte Grafikziel zeichnen	xxx
683	GRAPHIC GET BITS	eine Kopie einer Bitmap abrufen und als geräteunabhängige Bitmap in einer dynamischen String-Variable speichern	xxx
684	GRAPHIC GET CANVAS	die Puffergröße des angehängten Grafikziels abrufen	xxx
685	GRAPHIC GET CAPTION	die Beschriftung aus einem Grafikfenster abrufen	xxx
686	GRAPHIC GET CLIENT	die Client-Größe des ausgewählten Grafikziels abrufen	xxx
687	GRAPHIC GET CLIP	die Größe des Clip-Bereichs abrufen	xxx
688	GRAPHIC GET DC	das Handle des DC (Gerätekontext) für das ausgewählte Grafikziel abrufen	xxx
689	GRAPHIC GET LINES	die Anzahl der Zeilen ermitteln, die auf dem Grafikziel gedruckt werden können	xxx
690	GRAPHIC GET LOC	die Position des ausgewählten Grafikziels auf dem Bildschirm abrufen	xxx
691	GRAPHIC GET MIX	den Farbmischmodus für das ausgewählte Grafikziel abrufen	xxx
692	GRAPHIC GET OVERLAP	den Status des GRAPHIC-Überlappungsmodus abrufen	xxx
693	GRAPHIC GET PIXEL	die Farbe des Pixels am angegebenen Punkt im ausgewählten Grafikziel abrufen	xxx
694	GRAPHIC GET POS	den POS (letzter referenzierter Punkt) durch eine grafische Darstellung abrufen	xxx

695	GRAPHIC GET PPI	die Auflösung des Anzeigegeräts in Punkten pro Zoll abrufen	xxx
696	GRAPHIC GET SCALE	die aktuellen Koordinatengrenzen für das Grafikziel abrufen	xxx
697	GRAPHIC GET SCROLLTEXT	den Status des GRAPHIC-ScrollText-Modus abrufen	xxx
698	GRAPHIC GET SIZE	die Gesamtgröße des ausgewählten Grafikziels abrufen.	xxx
699	GRAPHIC GET STRETCHMODE	den Standard-BitmapStretching-Modus für den angehängten DC abrufen	xxx
700	GRAPHIC GET VIEW	die Position des virtuellen GRAPHIC-Viewports abrufen	xxx
701	GRAPHIC GET WORDWRAP	den Status des grafischen Zeilenumbruchmodus abrufen	xxx
702	GRAPHIC GET WRAP	den Status des GRAPHIC-Umbruchmodus abrufen	xxx
703	GRAPHIC IMAGELIST	ein Bild aus einer IMAGELIST anzeigen	xxx
704	GRAPHIC INKEY\$	ein Tastaturzeichen lesen, wenn eines vom Grafikziel bereit ist	xxx
705	GRAPHIC INPUT	Daten von der Tastatur lesen aus einem Grafikfenster	xxx
706	GRAPHIC INSTAT	ermitteln, ob ein Tastaturzeichen bereit ist	xxx
707	GRAPHIC INPUT FLUSH	alle gepufferten Tastaturdaten entfernen	xxx
708	GRAPHIC LINE	eine Linie in das ausgewählte Grafikziel zeichnen	xxx
709	GRAPHIC LINE INPUT	eine ganze Zeile von der Tastatur lesen aus dem Grafikfenster	xxx
710	GRAPHIC PAINT	einen Bereich mit einer Volltonfarbe füllen oder einem Schraffurmuster	xxx
711	GRAPHIC PIE	einen Kreisabschnitt auf dem ausgewählten Grafikziel zeichnen	xxx
712	GRAPHIC POLYGON	ein Polygon im ausgewählten Grafikziel zeichnen	xxx
713	GRAPHIC POLYLINE	eine Reihe verbundener Liniensegmente zeichnen	xxx
714	GRAPHIC PRINT	Textausgabe an das ausgewählte Grafikziel	xxx
715	GRAPHIC REDRAW	die gepufferte grafische Anweisungen aktualisieren und in das ausgewählte Grafikziel zeichnen	xxx
716	GRAPHIC RENDER	ein Bild auf dem ausgewählten Grafikziel rendern (Erstellung einer Grafik aus Rohdaten)	xxx
717	GRAPHIC SAVE	ein Bild in einer Bitmap-Datei (.BMP) speichern	xxx
718	GRAPHIC SCALE	ein benutzerdefiniertes Koordinatensystem für das Grafikziel definieren	xxx
719	GRAPHIC SET AUTOSIZE	erweitert ein Grafikziel in den AutoSize-Modus	xxx
720	GRAPHIC SET BITS	eine Kopie einer Bitmap ersetzen, die als geräteunabhängige Bitmap abgerufen wurde	xxx
721	GRAPHIC SET CAPTION	die Beschriftung eines Grafikfensters ändern	xxx
722	GRAPHIC SET CLIENT	die Größe eines Grafiksteuerelements ändern oder das Grafikfenster auf eine bestimmte Clientbereichsgröße	xxx
723	GRAPHIC SET CLIP	die Ränder um die Außenkanten des Grafikziels festlegen	xxx
724	GRAPHIC SET FIXED	ein Grafikziel auf den Standard-Festmodus zurückstellen	xxx

725	GRAPHIC SET FOCUS	das ausgewählte Grafikfenster in den Vordergrund bringen und den Fokus darauf richten	xxx
726	GRAPHIC SET FONT	eine Schriftart für die Anweisungen “GRAPHIC PRINT“, “GRAPHIC INPUT“ und “GRAPHIC LINE INPUT“ auswählen	xxx
727	GRAPHIC SET LOC	die Position des ausgewählten Grafikfensters auf dem Bildschirm ändern	xxx
728	GRAPHIC SET MIX	den Farbmischmodus für das ausgewählte Grafikziel festlegen	xxx
729	GRAPHIC SET OVERLAP	den GRAPHIC-Überlappungsmodus aktivieren oder deaktivieren	xxx
730	GRAPHIC SET PIXEL	ein einzelnes Pixel in das ausgewählte Grafikfenster zeichnen	xxx
731	GRAPHIC SET POS	den letzten Referenzpunkt (POS) für das ausgewählte Grafikziel festlegen	xxx
732	GRAPHIC SET SCROLLTEXT	den grafischen ScrollText-Modus aktivieren oder deaktivieren	xxx
733	GRAPHIC SET SIZE	die Gesamtgröße eines Grafiksteuerelements oder Grafikfensters ändern	xxx
734	GRAPHIC SET STRETCHMODE	den Standard-Bitmap-Streckmodus für den aktuellen DC festlegen	xxx
735	GRAPHIC SET VIEW	die Position des Ansichtsfensters auf einem virtuellen Grafikziel ändern.	xxx
736	GRAPHIC SET VIRTUAL	ein Grafikziel in den virtuellen Modus erweitern	xxx
737	GRAPHIC SET WORDWRAP	den grafischen WORDWRAP-Modus aktivieren oder deaktivieren	xxx
738	GRAPHIC SET WRAP	den grafischen WRAP-Modus aktivieren oder deaktivieren	xxx
739	GRAPHIC SPLIT	eine Zeichenfolge zur Anzeige auf einem Grafikziel in zwei Teile aufteilen	xxx
740	GRAPHIC STRETCH	eine Bitmap auf das ausgewählte Grafikziel kopieren und skalieren	xxx
741	GRAPHIC STYLE	den Linienstil festlegen, der von verschiedenen grafischen Anweisungen im ausgewählten Grafikziel verwendet werden soll	xxx
742	GRAPHIC TEXT SIZE	die Größe des zu druckenden Textes berechnen	xxx
743	GRAPHIC WAITKEY\$	ein Tastaturzeichen aus dem Grafikfenster lesen und warten, bis eines bereit ist	xxx
744	GRAPHIC WIDTH	die Linienbreite festlegen, die von verschiedenen grafischen Anweisungen im ausgewählten Grafikziel verwendet werden soll	xxx
745	GRAPHIC WINDOW	ein neues Grafikfenster erstellen	xxx
746	GRAPHIC WINDOW CLICK	prüfen, ob mit der Maus auf ein Grafikfenster geklickt wurde	xxx
747	GRAPHIC WINDOW END	das ausgewählte Grafikfenster schließen und zerstören	xxx
748	GRAPHIC WINDOW HIDE	ein Grafikfenster unsichtbar machen	xxx
749	GRAPHIC WINDOW MINIMIZE	ein Grafikfenster minimieren	xxx
750	GRAPHIC WINDOW NONSTABLE	ein Grafikfenster instabil (schließbar) machen	xxx
751	GRAPHIC WINDOW NORMALIZE	ein Grafikfenster sichtbar machen	xxx

752	GRAPHIC WINDOW STABILIZE	ein Grafikfenster stabil machen (nicht schließbar)	xxx
753	GRAPHIC WINDOW TEXT	ein neues Grafikfenster erstellen, das stärker auf die Anzeige von Text ausgerichtet ist	xxx
754	IMAGELIST ADD BITMAP	ein Bitmap-Bild wird zur IMAGELIST hinzugefügt	xxx
755	IMAGELIST ADD ICON	ein Symbolbild wird zur IMAGELIST hinzugefügt	xxx
756	IMAGELIST ADD MASKED	eine Bitmap wird der ICON IMAGELIST hinzugefügt	xxx
757	IMAGELIST GET COUNT	die Anzahl der Bilder in der IMAGELIST wird abgerufen	xxx
758	IMAGELIST KILL	die angegebene IMAGELIST wird zerstört	xxx
759	IMAGELIST NEW BITMAP	eine neue Bitmap-IMAGELIST-Struktur wird erstellt	xxx
760	IMAGELIST NEW ICON	eine neue Symbol-IMAGELIST-Struktur wird erstellt	xxx
761	IMAGELIST SET OVERLAY	ein Bild angegeben, das als Overlay verwendet werden soll	xxx
762	RGB	gibt einen RGB-Farbwert zur Verwendung mit der Windows-API-Palette und GDIs zurück	xxx

11. Eingabekommandos

Die folgenden Funktionen können zum Erfassen von Eingabedaten verwendet werden:

LfdNr	Befehl	Beschreibung	Seite
763	COMM	den Wert oder Status eines Kommunikations- Parameters abrufen	xxx
764	COMM LINE	eine CR/LF-terminierte "Zeile" von Daten von einem seriellen Port empfangen	xxx
765	COMM RECV	von einem seriellen Port Binärdaten empfangen	xxx
766	COMMAND\$	gibt die zum Starten des Programms verwendete Befehlszeile zurück	xxx
767	ENVIRON	die Umgebungstabelle des aktuellen Programms ändern	xxx
768	ENVIRON\$	Zeichenfolgen aus der Umgebungstabelle des Betriebssystems abrufen	xxx
769	EOF	den Dateiende-Status einer Datei, seriellen oder TCP/UDP-Übertragung zurückgeben	xxx
770	FIELD	eine Feldzeichenfolge an einen Dateipuffer oder eine dynamische Zeichenfolgenvariable binden	xxx
771	FILESCAN	schnell eine INPUT- oder BINARY-Datei scannen, um Informationen zur Zeichenfolgenreihe zu erhalten	xxx
772	FREEFILE	gibt die nächste verfügbare <i>PowerBASIC</i> - Dateinummer zurück	xxx
773	GET	einen Datensatz aus einer Direktzugriffsdatei lesen	xxx
774	GET\$	eine Zeichenfolge aus einer im Binärmodus geöffneten Datei lesen	xxx

775	GRAPHIC INKEY\$	ein Tastaturzeichen einlesen, falls eines aus dem Grafikfenster bereitsteht	xxx
776	GRAPHIC INPUT	Daten von der Tastatur lesen aus einem Grafikfenster	xxx
777	GRAPHIC INPUT FLUSH	alle gepufferten Tastaturdaten entfernen	xxx
778	GRAPHIC INSTAT	ermitteln, ob ein Tastaturzeichen bereit ist	xxx
779	GRAPHIC LINE INPUT	eine ganze Zeile von der Tastatur lesen aus dem Grafikfenster	xxx
780	GRAPHIC WAITKEY\$	ein Tastaturzeichen aus dem Grafikfenster lesen und warten, bis eines bereit ist	xxx
781	GRAPHIC WINDOW CLICK	prüfen, ob mit der Maus ein Grafikfenster angeklickt wurde	xxx
782	INPUT#	Variable laden mit Daten aus einer sequentiellen Datei	xxx
783	INPUTBOX\$	INPUTBOX\$ zeigt ein Dialogfeld mit einer Eingabeaufforderung an	xxx
784	LINE INPUT#	Zeile(n) aus einer sequentiellen Datei in eine Zeichenfolgen-Variable oder ein Array einlesen	xxx
785	LOC	die aktuelle Suchposition in einer geöffneten Datei auf der Festplatte ermitteln	xxx
786	LOF	gibt die Länge (Größe) einer geöffneten Datenträger-Datei zurück	xxx
787	MSGBOX	eine MSGBOX anzeigen lassen mit der Benutzerauswahl OK/Cancel.	xxx
788	MSGBOX	eine MSGBOX anzeigen mit der Benutzer-Auswahl Cancel ("Abbrechen")	xxx
789	PEEK	gibt das Byte an einem bestimmten Speicherort zurück	xxx
790	PEEK\$	gibt eine Bytefolge zurück, die an einem bestimmten Speicherort beginnt	xxx

12. Speicherverwaltung

Die folgenden Funktionen manipulieren Speicher und dessen Zuweisung/Freigabe:

<u>LfdNr</u>	<u>Befehl</u>	<u>Beschreibung</u>	<u>Seite</u>
791	#REGISTER	die automatische Zuweisung von Registervariablen.steuern	xxx
792	#STACK	die maximal mögliche Stapelgröße festlegen	xxx
793	ASMDATA/END ASMDATA	einen Block definieren, in dem primitive schreibgeschützte Daten gespeichert werden	xxx
794	CODEPTR	eine 32-Bit-Adresse eines Labels, Subs oder einer Funktion erhalten	xxx
795	DIM	Deklarieren und Dimensionieren von Arrays, Skalarvariablen und Zeiger	xxx
796	ERASE	den Array-Speicher freigeben	xxx

797	GLOBAL	globale (gemeinsam genutzte) Variablen zwischen Subs, Funktionen, Methoden und Eigenschaften deklarieren	xxx
798	GLOBALMEM ALLOC	einen verschiebbaren Speicherblock einer bestimmten Größe zuweisen	xxx
799	GLOBALMEM FREE	die Zuordnung eines verschiebbaren Speicherblocks freigeben	xxx
800	GLOBALMEM LOCK	einen beweglichen Speicherblock an einem bestimmten Speicherplatz sperren	xxx
801	GLOBALMEM SIZE	die Größe eines verschiebbaren Speicherblocks abrufen	xxx
802	GLOBALMEM UNLOCK	einen beweglichen Speicherblock entsperren	xxx
803	INSTANCE	deklarieren von INSTANCE-Variablen, die für jedes Objekt eindeutig sind	xxx
804	LEN	gibt die logische Länge einer Variablen, eines UDT oder einer UNION zurück	xxx
805	LET	einer Variablen einen Wert zuweisen.	xxx
806	LET (with Objects)	eine Objektreferenz einer Objektvariablen zuweisen	xxx
807	LET (with Types)	einer benutzerdefinierten Typ-Variablen Daten zuweisen	xxx
808	LET (with Variants)	einer VARIANT-Variablen einen Wert oder eine Objektreferenz zuweisen	xxx
809	LOCAL	deklarieren von lokalen Variablen in einem Sub, einer Funktion, einer Methode oder einer Eigenschaft	xxx
810	MEMORY COPY	eine angegebene Anzahl von Bytes von einer Adresse zu einer anderen kopieren	xxx
811	MEMORY FILL	den Speicher mit einem angegebenen BYTE, WORD, DWORD oder Bytes aus einem Zeichenfolgen-Ausdruck füllen	xxx
812	MEMORY SWAP	der Speicher an der angegebenen Adresse wird mit den Daten an einer anderen Adresse ausgetauscht	xxx
813	PEEK	gibt das Byte an einem bestimmten Speicherort zurück	xxx
814	PEEK\$	gibt aufeinanderfolgende 1-Byte-Zeichen zurück, beginnend an einem bestimmten Speicherort	xxx
815	PEEK\$\$	gibt aufeinanderfolgende 2 Byte-Zeichen zurück, beginnend an einem bestimmten Speicherort	xxx
816	POKE	ein Byte an einem bestimmten Speicherort speichern	xxx
817	POKE\$	eine Bytefolge speichern, beginnend an einem bestimmten Speicherort	xxx
818	POKE\$\$	eine Sequenz von 2 Byte-Zeichen speichern, beginnend an einem bestimmten Speicherort	xxx
819	REDIM	deklarieren von dynamischen Arrays, Speicher zuweisen, neu zuweisen und die Zuordnung freigeben	xxx
820	REGISTER	definieren von lokalen Register-Variablen innerhalb einer Subfunktion, Funktion, Methode oder Eigenschaft	xxx

821	RESET	einen Array-Index oder ein ganzes Array auf Null oder Null / leer setzen.	xxx
822	SIZEOF	gibt die Gesamtlänge oder physikalische Länge einer beliebigen <i>PowerBASIC</i> -Variablen zurück	xxx
823	STATIC	deklarieren von statischen Variablen innerhalb einer Subfunktion, Funktion, Methode oder Eigenschaft	xxx
824	STRPTR	gibt die Adresse der Daten zurück, die in einer Zeichenfolge variabler Länge enthalten sind	xxx
825	THREADED	deklarieren von Thread-lokalen Variablen	xxx
826	TYPE/END TYPE	definieren eines benutzerdefinierten Typ und ein oder mehrere Mitglieds-Elemente	xxx
827	TYPE SET	einem UDT den Wert eines UDT oder eines Zeichenfolgen-Ausdrucks zuweisen	xxx
828	UNION/END UNION	erstellen einer UDT-ähnliche Struktur, deren Elemente sich im Speicher überlappen	xxx
829	VARPTR	gibt die 32-Bit-Adresse eines Variablen- oder String-Handles zurück	xxx

13. Meta-Anweisungen

Die folgenden Funktionen steuern das Compiler- und Debugger-Verhalten:

LfdNr	Befehl	Beschreibung	Seite
830	#ALIGN	die nächste Anweisung an einer Grenze ausrichten	xxx
831	%DEF	ermitteln, ob zuvor ein Equate definiert wurde	xxx
832	#COM DOC	gibt eine Hilfezeichenfolge an, die normalerweise eine allgemeine Beschreibung des COM-Servers enthält	xxx
833	#COM HELP	gibt den Namen der zugehörigen Hilfedatei und den Hilfekontext-Code an	xxx
834	#COM NAME	gibt den Namen des Servers und die Versionsnummer an	xxx
835	#COM GUID	gibt die GUID an, die die gesamte Anwendung oder Bibliothek identifiziert (APPID oder LIBID)	xxx
836	#BLOAT	künstlich die Größe des Disk-Images eines kompilierten Programms erhöhen	xxx
837	#COMPILE	bestimmen, welcher Dateityp vom Compiler erstellt werden soll	xxx
838	#COMPILER	definieren des Compilers für dieses Programm	xxx
839	#DEBUG CODE	Compiler-Direktive zum Unterdrücken der Generierung von Debug-Code	xxx
840	#DEBUG DISPLAY	zeigt eine Meldung an, wenn ein nicht abgefangener Laufzeitfehler auftritt	xxx
841	#DEBUG ERROR	steuert die Generierung von Fehlerprüfcode	xxx
842	#DEBUG PRINT	zeigt Informationen im Debug-Fenster der IDE an	xxx
843	#DIM	Angabe, ob Variablen vor der Verwendung deklariert werden müssen	xxx

844	#EXPORT	deklarieren einer Unterfunktion/Funktion, die das Attribut EXPORT hat	xxx
845	#IF	definieren von Abschnitten des Quellcodes, die kompiliert oder ignoriert werden sollen	xxx
846	#INCLUDE	den Compiler anweisen, eine zusätzliche Quelldatei von der Festplatte zu lesen	xxx
847	#LINK	eine vorkompilierte Static Link Library (SLL) mit dem Hostprogramm verknüpfen	xxx
848	#MESSAGES	Angabe, welche Nachrichten an eine Control Callback-Funktion gesendet werden sollen	xxx
849	#OPTIMIZE	die Optimierung auswählen, die auf das Programm angewendet werden soll	xxx
850	#OPTION	verschiedene Compileroptionen festlegen	xxx
851	#PAGE	eine Seitengrenze für die <i>PowerBASIC</i> -IDE festlegen	xxx
852	#PBFORMS	Anweisungen für den <i>Visual Designer</i> von <i>PowerBASIC Forms</i>	xxx
853	#REGISTER	steuert die automatische Zuweisung von Register-Variablen	xxx
854	#RESOURCE	eine <i>PowerBASIC</i> -Ressourcen-Datei in die ausführbare Datei einbetten	xxx
855	#STACK	die maximal mögliche Stapelgröße festlegen	xxx
856	#TOOLS	aktivieren/deaktivieren von integrierten Entwicklungstools im kompilierten Code	xxx
857	#UNIQUE	Angabe, ob eindeutige Variablennamen erforderlich sind	xxx
858	#UTILITY	Compiler-Direktive, die es externen Dienstprogrammen ermöglicht, in die Zeile #UTILITY eingefügten Text zu lesen	xxx

14. Numerische Operationen

Die folgenden Funktionen manipulieren und verwalten numerische Daten:

LfdNr	Befehl	Beschreibung	Seite
859	ABS	gibt den Absolutwert eines numerischen Ausdrucks zurück	xxx
860	AND	AND funktioniert sowohl als logischer als auch als bitweise arithmetischer Operator	xxx
861	ARRAY ASSIGN	aufeinanderfolgenden Elementen eines Arrays eine Reihe von Werten zuweisen	xxx
862	ARRAY DELETE	ein einzelnes Element aus einem bestimmten Array löschen	xxx
863	ARRAY INSERT	ein einzelnes Element in ein gegebenes Array einfügen	xxx
864	ARRAY SCAN	das gesamte oder einen Teil eines Arrays nach einem bestimmten Wert scannen	xxx
865	ARRAY SORT	das gesamte oder einen Teil eines gegebenen Arrays sortieren	xxx

866	ASC	gibt den ASCII-Code des angegebenen Zeichens in einer Zeichenfolge zurück	xxx
867	ASC	ein ASCII-Byte an der angegebenen Position in einer Zeichenfolge einfügen	xxx
868	ATN	gibt den Arkustangens seines Arguments zurück	xxx
869	BIN\$	gibt eine Zeichenfolge mit der binären (Basis 2) -Darstellung eines Wertes zurück	xxx
870	BIT CALC	ein Bit in einer Variablen der INTEGER-CLASS setzen oder zurücksetzen	xxx
871	BIT	gibt den Wert eines bestimmten Bits in einer Variablen der INTEGRAL-CLASS zurück	xxx
872	BIT	einzelne Bits einer Variablen der INTEGRAL-CLASS manipulieren	xxx
873	BITS	gibt den am wenigsten signifikanten Teil eines INTEGRAL-CLASS Wertes zurück	xxx
874	BITS	gibt die niedrigstwertigen 8, 16 oder 32 Bits eines Arguments zurück	xxx
875	BITSE	vergleicht die ganzzahlige Werte für äquivalente Bits unabhängig vom Vorzeichen	xxx
876	CBYT	konvertiert einen Wert in einen Byte-Datentyp	xxx
877	CCUR	konvertiert einen Wert in den Datentyp "Währung"	xxx
878	CCUX	konvertiert einen Wert in einen erweiterten Datentyp "Währung"	xxx
879	CDBL	konvertiert einen Wert in einen Datentyp mit doppelter Genauigkeit	xxx
880	CDWD	Konvertieren eines Wertes in einen DoubleWord-Datentyp	xxx
881	CEIL	gibt einen Wert zurück, der größer oder gleich einem Argument ist	xxx
882	CEXT	konvertiert einen Wert in einen Datentyp mit erweiterter Genauigkeit	xxx
883	CHOOSE	gibt einen von mehreren Werten zurück, basierend auf dem Wert eines Index	xxx
884	CINT	konvertiert einen Wert in einen ganzzahligen Datentyp	xxx
885	CLNG	konvertiert einen Wert in einen LongInteger-Datentyp	xxx
886	COS	gibt den Kosinus eines Arguments zurück	xxx
887	CQUD	konvertiert einen Wert in einen QuadInteger-Datentyp	xxx
888	CSNG	konvertiert einen Wert in einen Datentyp mit einfacher Genauigkeit	xxx
889	CVBYT	konvertiert binär codierter Zeichenfolgen-Daten in einen Bytewert	xxx
890	CVCUR	konvertieren binär codierter Zeichenfolgen-Daten in einen Währungswert	xxx
891	CVCUX	konvertiert binär codierte Zeichenfolgendaten in erweiterte Währung	xxx

892	CVD	konvertiert binär codierte Zeichenfolgendaten in einen Wert mit doppelter Genauigkeit	xxx
893	CVDWD	konvertiert binär codierte Zeichenfolgendaten in einen DoubleWord-Wert	xxx
894	CVE	konvertiert binär codierte Zeichenfolgendaten in Daten mit erweiterter Genauigkeit	xxx
895	CVI	konvertiert binär codierte Zeichenfolgendaten in einen ganzzahligen Wert	xxx
896	CVL	konvertiert binär codierte Zeichenfolgendaten in einen LongInteger-Wert	xxx
897	CVQ	konvertiert binär codierte Zeichenfolgendaten in einen QuadInteger-Wert	xxx
898	CVS	konvertiert binär codierte Zeichenfolgendaten in einen Wert mit einfacher Genauigkeit	xxx
899	CVWRD	konvertiert binär codierte Zeichenfolgendaten in einen word-Wert	xxx
900	CWRD	konvertiert einen Wert in einen word-Datentyp	xxx
901	DEC\$	konvertiert einen ganzzahligen Wert in eine Dezimal-Zeichenfolge.	xxx
902	DECR	eine Variable, ein Zeiger oder eines Zeigerziels wird dekrementiert	xxx
903	DEFBYT	den Standard-Variablentyp als Byte deklarieren	xxx
904	DEFCUR	den Standard-Variablentyp als Währung deklarieren	xxx
905	DEFCUX	den Standard-Variablentyp als "Erweiterte Währung" deklarieren	xxx
906	DEFDBL	den Standard-Variablentyp als doppelte Genauigkeit deklarieren	xxx
907	DEFDWD	den Standard-Variablentyp als DoubleWord deklarieren	xxx
908	DEFEXT	den Standard-Variablentyp als erweiterte Genauigkeit deklarieren	xxx
909	DEFINT	den Standard-Variablentyp als ganzzahligen Wert deklarieren	xxx
910	DEFLNG	den Standard-Variablentyp als Long-Integer deklarieren	xxx
911	DEFQUD	den Standardvariablentyp als Quad-Integer deklarieren	xxx
912	DEFSNG	den Standard-Variablentyp als Single-Precision deklarieren	xxx
913	DEFSTR	den Standard-Variablentyp als String deklarieren	xxx
914	DEFWRD	den Standard-Variablentyp als word deklarieren	xxx
915	ENUM/END ENUM	eine Gruppe logisch verknüpfter numerischer Gleichungen erstellen	xxx
916	EQV	eine logische oder bitweise Äquivalenz-Operation durchführen	xxx
917	EXP	gibt eine Basiszahl potenziert mit einer Basis von "e" zurück	xxx
918	EXP2	gibt eine Basiszahl potenziert mit der Basis 2 zurück	xxx

919	EXP10	gibt eine Basiszahl potenziert mit der Basis 10 zurück	xxx
920	FIX	eine Gleitkommazahl auf einen ganzzahligen Wert einkürzen	xxx
921	FORMAT\$	numerische Daten gemäß einem Zeichenfolgen-Maskenausdruck formatieren	xxx
922	FRAC	gibt den Bruchteil einer Gleitkommazahl zurück	xxx
923	HEX\$	hexadezimale (Basis 16) Zeichenfolgen-Darstellung eines Arguments	xxx
924	HI	den signifikantesten (höchstwertigen) Teil eines Arguments extrahieren	xxx
925	IIF	gibt einen von zwei Werten basierend auf einer True/False-Auswertung zurück	xxx
926	IMP	eine logische oder bitweise Implication-Operation durchführen	xxx
927	INCR	eine Variable, einen Zeiger oder ein Zeigerziel inkrementieren	xxx
928	INT	einen numerischen Ausdruck in einen Wert des Ganzzahl-Formats konvertieren	xxx
929	ISFALSE	gibt die logische Falschheit eines gegebenen Ausdrucks zurück	xxx
930	ISNOTHING	den aktuellen Status einer bestimmten Objektvariablen ermitteln	xxx
931	ISOBJECT	den aktuellen Status einer bestimmten Objektvariablen ermitteln	xxx
932	ISTRUE	gibt die logische Wahrheit eines gegebenen Ausdrucks zurück	xxx
933	LBOUND	gibt den niedrigsten Index der spezifischen Dimension eines Arrays zurück	xxx
934	LEN	gibt die logische Länge (Grösse) einer Variablen, eines UDT oder einer Union zurück	xxx
935	LET	einer Variablen einen Wert zuweisen	xxx
936	LET (with Variants)	einer Variant-Variablen einen Wert oder eine Objektreferenz zuweisen	xxx
937	LO	den am wenigsten signifikanten (niedrigstwertigen) Teil eines Arguments extrahieren	xxx
938	LOG	gibt den natürlichen Logarithmus (Basis e) eines Arguments zurück	xxx
939	LOG2	gibt den Logarithmus zur Basis 2 eines Arguments zurück	xxx
940	LOG10	gibt den Logarithmus zur Basis 10 eines Arguments zurück	xxx
941	MAT	Matrixberechnungen auf numerischen Arrays	xxx
942	MAX	gibt das Argument mit dem größten (maximalen) Wert zurück	xxx
943	MIN	gibt das Argument mit dem kleinsten (minimalen) Wert zurück	xxx
944	MOD	gibt den Rest der Division zwischen zwei Zahlen zurück	xxx

945	NOT	der NOT-Operator funktioniert als bitweiser arithmetischer Operator	xxx
946	OCT\$	gibt eine Zeichenfolge zurück, die eine oktale (Basis 8) Darstellung eines Werts ist	xxx
947	OR	eine logische oder bitweise ODER-Rechenoperation durchführen	xxx
948	PEEK	gibt das Byte an einem bestimmten Speicherort zurück	xxx
949	POKE	ein Byte an einem bestimmten Speicherort speichern	xxx
950	RANDOMIZE	den Zufallszahlen-Generator starten	xxx
951	RESET	eine Variable, einen Array-Index oder ein ganzes Array auf Null setzen	xxx
952	RGB	gibt einen zusammengesetzten RGB-Farbwert zurück	xxx
953	RND	gibt eine Zufallszahl zurück	xxx
954	ROTATE	die Bits in einer Variable der Integralklasse rotieren	xxx
955	ROUND	einen numerischen Wert auf eine angegebene Anzahl von Dezimalstellen runden	xxx
956	SGN	gibt das Vorzeichen eines numerischen Ausdrucks zurück	xxx
957	SHIFT	die Bits in einer Variable der Integralklasse verschieben	xxx
958	SIN	gibt den Sinus eines Arguments zurück	xxx
959	SQR	gibt die Quadratwurzel eines Arguments zurück	xxx
960	SWAP	die Werte zweier Variablen, Zeiger oder Zeigerziele austauschen	xxx
961	SWITCH	gibt ein Element einer Reihe basierend auf einer True/False-Auswertung zurück	xxx
962	TAN	gibt den Tangens eines Arguments zurück	xxx
963	UBOUND	gibt den höchsten Index der spezifischen Dimension eines Arrays zurück	xxx
964	USING\$	Zeichenfolgen-/numerische Ausdrücke mithilfe einer Maskenzeichenfolge formatieren	xxx
965	VAL function	gibt das numerische Äquivalent eines String-Arguments zurück	xxx
966	VAL statement	konvertiert eine Textzeichenfolge in einen numerischen Wert mit zusätzlichen Informationen.	xxx
967	VARIANT#	gibt den numerischen Wert zurück, der in einer Variant-Variablen enthalten ist	xxx
968	XOR	eine logische oder bitweise Exklusiv-ODER-Operation durchführen	xxx

Die folgenden Funktionen manipulieren Datei- und Betriebssystemfunktionen:

LfdNr	Befehl	Beschreibung	Seite
969	CHDIR	ändert das aktuelle (Standard-)Verzeichnis auf einem bestimmten Laufwerk	xxx
970	CHDRIVE	ändert das aktuelle Standardlaufwerk	xxx
971	CLIPBOARD GET BITMAP	eine Bitmap wird aus der Zwischenablage kopiert und in einer neu erstellten GRAFIK-BITMAP gespeichert	xxx
972	CLIPBOARD GET OEMTEXT	eine Textzeichenfolge wird aus der Zwischenablage abgerufen und bei Bedarf in das OEM-Textformat konvertiert	xxx
973	CLIPBOARD GET TEXT	eine Textzeichenfolge wird aus der Zwischenablage abgerufen und bei Bedarf in das ASCII-Textformat konvertiert	xxx
974	CLIPBOARD GET UNICODE	eine Textzeichenfolge wird aus der Zwischenablage abgerufen und bei Bedarf in das Unicode-Textformat konvertiert	xxx
975	CLIPBOARD RESET	der Inhalt der Zwischenablage wird gelöscht	xxx
976	CLIPBOARD SET BITMAP	kopiert eine GRAFIK BITMAP in die Zwischenablage	xxx
977	CLIPBOARD SET OEMTEXT	kopiert eine OEM-Textzeichenfolge in die Zwischenablage	
978	CLIPBOARD SET TEXT	kopiert eine ASCII-Textzeichenfolge in die Zwischenablage	xxx
979	CLIPBOARD SET UNICODE	kopiert eine Unicode-Textzeichenfolge in die Zwischenablage	xxx
980	COMMAND\$	gibt die zum Starten des Programms verwendete Befehlszeile zurück	xxx
981	CURDIR\$	gibt das aktuelle Verzeichnis für ein bestimmtes Laufwerk zurück	xxx
982	DATE\$	das Systemdatum festlegen oder abrufen	xxx
983	DESKTOP GET CLIENT	die Größe des Clientbereichs des Desktops in Pixeln abrufen	xxx
984	DESKTOP GET LOC	die Position der oberen linken Ecke des Clientbereichs des Desktops in Pixeln abrufen	xxx
985	DESKTOP GET SIZE	die Größe des gesamten Desktops in Pixeln abrufen	xxx
986	DIR\$	gibt einen Dateinamen zurück, der mit der angegebenen Maske übereinstimmt	xxx
987	DIR\$ CLOSE	die Freigabe des FindNext-Handles des Betriebssystems erzwingen	xxx
988	DISKFREE	gibt die Größe des verfügbaren Speicherplatzes einer Festplatte in Bytes zurück	xxx
990	DISKSIZE	gibt den Gesamtspeicherplatz auf einer Festplatte in Bytes zurück	xxx
991	DISPLAY BROWSE	einen Ordnerauswahl-Dialog anzeigen, um die Auswahl des Benutzers zurückzugeben	xxx
992	DISPLAY COLOR	einen Farbauswahl-Dialog anzeigen, um die Auswahl des Benutzers zurückzugeben	xxx
993	DISPLAY FONT	einen Auswahldialog anzeigen, um die Benutzerauswahl zurückzugeben	xxx

994	DISPLAY OPENFILE	einen OpenFileDialog anzeigen, um die Benutzerauswahl zurückzugeben	xxx
995	DISPLAY SAVEFILE	einen SaveFile-Auswahldialog anzeigen, um die Benutzerauswahl zurückzugeben	xxx
996	ENVIRON	die Umgebungstabelle des aktuellen Programms ändern	xxx
997	ENVIRON\$	Zeichenfolgen aus der Umgebungstabelle des Betriebssystems abrufen	xxx
998	EXE.Inst	gibt das Instanz-Handle der aktuell ausgeführten Programmierung zurück	xxx
999	EXE.Exttn\$	gibt die Erweiterung des aktuell ausgeführten Programms zurück	xxx
1000	EXE.Full\$	gibt vollständig das Laufwerk, den Pfad und den Dateinamen des aktuell ausgeführten Programms zurück	xxx
1001	EXE.Name\$	gibt nur den Dateinamen des aktuell ausgeführten Programms zurück	xxx
1002	EXE.Namex\$	gibt den Dateinamen und die Erweiterung des aktuell ausgeführten Programms zurück	xxx
1003	EXE.Path\$	gibt vollständig das Laufwerk und den Pfad des aktuell ausgeführten Programms zurück	xxx
1004	FILEATTR	gibt Informationen einer geöffneten Datei zurück	xxx
1005	FILECOPY	kopiert eine Datei	xxx
1006	FILENAME\$	gibt den Dateisystem-Namen einer geöffneten Datei zurück	xxx
1007	FLUSH	die Dateipuffer auf die Festplatte leeren, um sicherzustellen, dass die Festplatten-Informationen aktuell sind	xxx
1008	GETATTR	gibt die Dateisystem-Attribute einer Datenträgerdatei oder eines Verzeichnisses zurück	xxx
1009	HOST ADDR	einen Hostnamen in eine entsprechende IP-Adresse konvertieren	xxx
1010	HOST NAME	eine IP-Adresse in einen entsprechenden Hostnamen konvertieren.	xxx
1011	ISFILE	feststellen, ob eine Datei vorhanden ist oder nicht	xxx
1012	KILL	eine Datenträgerdatei löschen	xxx
1013	METRICS	Informationen oder Dimensionen von Systemelementen abrufen	xxx
1014	MKDIR	ein Unterverzeichnis/einen Ordner erstellen (wie mit dem DOS-Befehl MKDIR)	xxx
1015	NAME	eine Datei oder ein Verzeichnis umbenennen (wie der DOS-Befehl REN)	xxx
1016	OPEN	eine Datei oder ein Gerät zum Lesen oder Schreiben vorbereiten	xxx
1017	PATHNAME\$	einen Pfad-/Dateinamen analysieren, um Komponententeile zu extrahieren.	xxx
1018	PATHSCAN\$	eine Datei auf der Festplatte suchen und den Pfad und/oder Dateinamenteile zurückgeben	xxx
1019	RGB	gibt einen zusammengesetzten RGB-Farbwert zurück	xxx

1020	RMDIR	ein Datenträgerverzeichnis löschen (wie mit dem DOS-Befehl RMDIR).	xxx
1021	SETATTR	die Dateisystem-Attribute einer Datenträgerdatei oder eines Verzeichnisses festlegen	xxx
1022	SETEOF	ein Datei bis zur aktuellen Dateizeiger-Position kürzen / erweitern.	xxx
1023	SHELL	ein ausführbares Programm synchron / asynchron starten	xxx
1024	SLEEP	den aktuellen Thread für eine angegebene Anzahl von Millisekunden pausieren	xxx

16. Druckbefehle

Die folgenden Befehle werden zum Senden von Daten an einen Drucker verwendet:

LfdNr	Befehl	Beschreibung	Seite
1025	LPRINT	Text und Daten an ein Druckergerät ausgeben	xxx
1026	LPRINT ATTACH	direkte Verbindung mit einem Zeilendrucker herstellen	xxx
1027	LPRINT CLOSE	das aktuelle Druckergerät von der Auagabe trennen	xxx
1028	LPRINT FLUSH	alle verbleibenden Druckdaten auf das Druckergerät übertragen	xxx
1029	LPRINT FORMFEED	ein Formfeed-Zeichen (Seitenauswurf) an den Drucker senden	xxx
1030	LPRINT\$	gibt das aktuelle Druckergerät zurück, das für LPRINT-Vorgänge verwendet wird	xxx
1031	PRINTER\$	Drucker-Namen und Druckeranschluss-Namen abrufen	xxx
1032	PRINTERCOUNT	die Anzahl der verfügbaren (installierten) Drucker abrufen	xxx
1033	XPRINT(CANVAS.X)	die beschreibbare Breite der Host-Druckerseite abrufen	xxx
1034	XPRINT(CANVAS.Y)	die beschreibbare Höhe der Hostdruckerseite abrufen	xxx
1035	XPRINT(Cell.Size.X)	die Zeichenzellen-Breite einschließlich des äußeren Zeilenabstands abrufen	xxx
1036	XPRINT(Cell.Size.Y)	die Zeichenzellen-Höhe einschließlich des externen Zeilenabstands abrufen	xxx
1037	XPRINT(Chr.Size.X)	die Zeichen-Breite auf der Hostdruckerseite abrufen	xxx
1038	XPRINT(Chr.Size.Y)	die Zeichen-Höhe auf der Hostdruckerseite abrufen	xxx
1039	XPRINT(Client.X)	die Breite des Clientbereichs (druckbarer Bereich) auf der Hostdruckerseite abrufen	xxx
1040	XPRINT(Client.Y)	die Höhe des Clientbereichs (druckbarer Bereich) auf der Hostdruckerseite abrufen	xxx
1041	XPRINT(Clip.X)	die Breite des Clipbereichs auf dem ausgewählten Drucker abrufen	xxx
1042	XPRINT(Clip.Y)	die Höhe des Clipbereichs auf dem ausgewählten Drucker abrufen	xxx

1043	XPRINT(COL)	die nächste Spaltendruckposition basierend auf der Zeilen- und Spaltenposition einer Textzelle abrufen	xxx
1044	XPRINT(COLLATE)	den XPRINT-Sortierungsstatus abrufen	xxx
1045	XPRINT(COLORMODE)	den Status vom XPRINT-Farbmodus abrufen	xxx
1046	XPRINT(COPIES)	die XPRINT-Kopienanzahl abrufen	xxx
1047	XPRINT(DC)	das Handle des Gerätekontexts (DC) für die Host-Druckerseite abrufen	xxx
1048	XPRINT(DUPLEX)	den XPRINT-Duplexstatus abrufen	xxx
1049	XPRINT(LINES)	die Anzahl der Zeilen abrufen, die gedruckt werden können.	xxx
1050	XPRINT(MIX)	den Farbmischmodus für eine Hostdruckerseite abrufen	xxx
1051	XPRINT(ORIENTATION)	die Papierausrichtung für eine Hostdruckerseite abrufen	xxx
1052	XPRINT(OVERLAP)	den Status des XPRINT-Überlappungsmodus abrufen	xxx
1053	XPRINT(PAPER)	die aktuelle Papiergröße / den aktuellen Papiertyp abrufen	xxx
1054	XPRINT(PIXEL...)	die Farbe eines Pixels auf einer Hostdruckerseite abrufen	xxx
1055	XPRINT(POS.X)	den letzten horizontalen Punkt abrufen, auf den durch eine XPRINT-Anweisung verwiesen wird (POS)	xxx
1056	XPRINT(POS.Y)	den letzten vertikalen Punkt abrufen, auf den durch eine XPRINT-Anweisung verwiesen wird (POS)	xxx
1057	XPRINT(PPI.X)	die horizontale Auflösung der Hostdruckerseite abrufen	xxx
1058	XPRINT(PPI.Y)	die vertikale Auflösung der Hostdruckerseite abrufen	xxx
1059	XPRINT(QUALITY)	die Einstellung der Druckqualität für den Hostdrucker abrufen	xxx
1060	XPRINT(ROW)	die nächste Zeilendruckposition basierend auf der Zeilen- und Spaltenposition einer Textzelle abrufen	xxx
1061	XPRINT(SELECTION)	den Status des SELECTION-Flags abrufen	xxx
1062	XPRINT(SIZE.X)	die Breite der Host-Druckerseite abrufen	xxx
1063	XPRINT(SIZE.Y)	die Höhe der Hostdruckerseite abrufen	xxx
1064	XPRINT(STRETCHMODE)	den Standard-BitmapStretch-Modus für den angehängten DC abrufen	xxx
1065	XPRINT(TEXT.SIZE.X...)	die Breite des auf einem Hostdrucker zu druckenden Textes berechnen	xxx
1066	XPRINT(TEXT.SIZE.Y...)	die Höhe des auf einem Hostdrucker zu druckenden Textes berechnen	xxx
1067	XPRINT(TRAY)	das aktive Druckerfach abrufen	xxx
1068	XPRINT(WORDWRAP)	den Status des XPRINT-wordWrap-Modus abrufen	xxx
1069	XPRINT(WRAP)	den Status des XPRINT wrap-Modus abrufen	xxx
1070	XPRINT\$	gibt den Namen des angeschlossenen Hostdruckers zurück	xxx
1071	XPRINT\$(ATTACH)	gibt den Namen des angeschlossenen Hostdruckers zurück	xxx
1072	XPRINT\$(PAPERS)	eine Liste der unterstützten Papiertypen abrufen	xxx
1073	XPRINT\$(TRAYS)	eine Liste der unterstützten Papierfächer abrufen	xxx

1074	XPRINT	Text an ein Host-Druckergerät ausgeben	XXX
1075	XPRINT ARC	einen Bogen auf einer Hostdruckerseite zeichnen	XXX
1076	XPRINT ATTACH	einen hostbasierten (GDI) Drucker zur Verwendung mit XPRINT anschließen	XXX
1077	XPRINT BOX	auf einer Hostdruckerseite ein Kästchen zeichnen mit quadratischen oder abgerundeten Ecken	XXX
1078	XPRINT CANCEL	einen Druckauftrag auf dem Hostdrucker abbrechen	XXX
1079	XPRINT CELL	die nächste Druckposition für eine Textzelle festlegen oder sie abrufen	XXX
1080	XPRINT CELL SIZE	die Größe der Zeichenzellen einschließlich des externen Zeilenabstands abrufen	XXX
1081	XPRINT CHR SIZE	die Zeichengröße für die aktuelle Schriftart auf einer Hostdruckerseite abrufen	XXX
1082	XPRINT CLOSE	einen Hostdrucker abtrennen, damit der Druckvorgang beginnen kann	XXX
1083	XPRINT COLOR	die Vordergrundfarbe (und optional die Hintergrundfarbe) für verschiedene XPRINT-Anweisungen festlegen	XXX
1084	XPRINT COPY	eine Bitmap auf eine Hostdruckerseite kopieren	XXX
1085	XPRINT ELLIPSE	eine Ellipse oder einen Kreis auf einer Hostdruckerseite zeichnen	XXX
1086	XPRINT FORMFEED	eine neue Seite für den Hostdrucker starten	XXX
1087	XPRINT GET ATTACH	den Namen des angeschlossenen Hostdruckers abrufen	XXX
1088	XPRINT GET CANVAS	die Puffergröße des angeschlossenen Hostdruckers abrufen	XXX
1089	XPRINT GET CLIENT	die Größe des Clientbereichs (druckbarer Bereich) auf der Hostdruckerseite abrufen	XXX
1090	XPRINT GET CLIP	die Größe des Clipbereichs auf dem ausgewählten Drucker abrufen	XXX
1091	XPRINT GET COLLATE	den XPRINT-Sortierstatus abrufen	XXX
1092	XPRINT GET COLORMODE	den Status des XPRINT-Farbmodus abrufen	XXX
1093	XPRINT GET COPIES	die Anzahl der XPRINT-Kopien abrufen	XXX
1094	XPRINT GET DC	das Handle des Gerätekontexts (DC) für die Hostdruckerseite abrufen	XXX
1095	XPRINT GET DUPLEX	den XPRINT-Duplexstatus abrufen	XXX
1096	XPRINT GET LINES	die Anzahl der Zeilen abrufen, die gedruckt werden können	XXX
1097	XPRINT GET MARGIN	die Randgrößen für den Hostdrucker abrufen	XXX
1098	XPRINT GET MIX	den Farbmischmodus für eine Hostdruckerseite abrufen	XXX
1099	XPRINT GET ORIENTATION	die Papierausrichtung für eine Hostdruckerseite abrufen	XXX
1100	XPRINT GET OVERLAP	den Status des XPrint-OVERLAP Modus abrufen	XXX
1101	XPRINT GET PAGES	die XPRINT-Seitenzahlbeschränkungen für diesen Druckauftrag abrufen	XXX
1102	XPRINT GET PAPER	aktuelles Papierformat / -typ abrufen	XXX
1103	XPRINT GET PAPERS	eine Liste der unterstützten Papiertypen abrufen	XXX
1104	XPRINT GET PIXEL	die Farbe eines Pixels auf einer Hostdruckerseite abrufen	XXX

1105	XPRINT GET POS	den letzten durch eine XPRINT-Anweisung referenzierten Punkt (POS) abrufen	XXX
1106	XPRINT GET PPI	die Auflösung der Hostdruckerseite abrufen	XXX
1107	XPRINT GET QUALITY	die Druckqualitäts-Einstellung für den Hostdrucker abrufen	XXX
1108	XPRINT GET SCALE	die aktuellen Koordinatengrenzen für die Hostdruckerseite abrufen	XXX
1109	XPRINT GET SELECTION	den Status des SELECTION-Flags abrufen	XXX
1110	XPRINT GET SIZE	die Gesamtgröße der Hostdruckerseite abrufen	XXX
1111	XPRINT GET STRETCHMODE	den Standard-Bitmap-Stretching-Modus für den angehängten DC abrufen	XXX
1112	XPRINT GET TRAY	das aktive Druckerfach abrufen	XXX
1113	XPRINT GET TRAYS	eine Liste der unterstützten Papierfächer abrufen	XXX
1114	XPRINT GET WORDWRAP	den Status des XPRINT wordWrap-Modus abrufen	XXX
1115	XPRINT GET WRAP	den Status des XPRINT wrap-Modus abrufen	XXX
1116	XPRINT IMAGELIST	ein Bild aus einer IMAGELIST drucken	XXX
1117	XPRINT LINE	eine Linie auf einer Hostdruckerseite zeichnen	XXX
1118	XPRINT PIE	einen Kreisabschnitt auf einer Hostdruckerseite zeichnen	XXX
1119	XPRINT POLYGON	ein Polygon auf einer Hostdruckerseite zeichnen	XXX
1120	XPRINT POLYLINE	eine Reihe verbundener Linien auf einer Hostdruckerseite zeichnen	XXX
1121	XPRINT PREVIEW	eine Kopie eines zu druckenden Dokuments auf dem Bildschirm anzeigen	XXX
1122	XPRINT PREVIEW CLOSE	die XPRINT-Ausgabe zurücksetzen zum Hostdrucker	XXX
1123	XPRINT PRINT	Text ausgeben, der auf dem ausgewählten Drucker gedruckt werden soll	XXX
1124	XPRINT RENDER	Rendern eines Bilds auf einer Hostdruckerseite	XXX
1125	XPRINT SCALE	ein benutzerdefiniertes Weltkoordinatensystem für eine Hostdruckerseite definieren	XXX
1126	XPRINT SCALE PIXELS	das Koordinatensystem auf die ursprünglichen Standardpixel-Koordinaten zurücksetzen	XXX
1127	XPRINT SET CLIP	die Ränder um die Außenkanten der Druckseite festlegen	XXX
1128	XPRINT SET COLLATE	den XPRINT-Sortierstatus ändern	XXX
1129	XPRINT SET COLORMODE	den XPRINT-Farbmodusstatus ändern	XXX
1130	XPRINT SET COPIES	die XPRINT-Kopienanzahl ändern	XXX
1131	XPRINT SET DUPLEX	den XPRINT-Duplexstatus ändern	XXX
1132	XPRINT SET FONT	eine Schriftart für die XPRINT-Anweisung auswählen	XXX
1133	XPRINT SET MIX	den Farbmischmodus für eine Hostdruckerseite festlegen	XXX
1134	XPRINT SET ORIENTATION	die Papierausrichtung für eine Hostdruckerseite festlegen	XXX
1135	XPRINT SET OVERLAP	aktiviert oder deaktiviert den XPRINT-Überlappungsmodus	XXX
1136	XPRINT SET PAGES	die XPRINT-Seitenzahlbegrenzungen für diesen Druckauftrag festlegen	XXX

1137	XPRINT SET PAPER	ein neues Papierformat / einen neuen Papiertyp festlegen	xxx
1138	XPRINT SET PIXEL	die Farbe eines Pixels auf einer Hostdruckerseite festlegen	xxx
1139	XPRINT SET POS	den letzten durch eine XPRINT-Anweisung referenzierten Punkt (POS) abrufen	xxx
1140	XPRINT SET QUALITY	die Druckqualität für einen Hostdrucker festlegen	xxx
1141	XPRINT SET STRETCHMODE	den Standard-BitmapStreck-Modus für den aktuellen DC festlegen	xxx
1142	XPRINT SET TRAY	ein neues aktives Druckerfach festlegen	xxx
1143	XPRINT SET WORDWRAP	aktiviert oder deaktiviert den XPRINT-wordwrap-Modus	xxx
1144	XPRINT SET WRAP	aktiviert oder deaktiviert den XPRINT-wrap-Modus	xxx
1145	XPRINT SPLIT	teilt eine Zeichenfolge zum Drucken mit XPRINT in zwei Teile auf	xxx
1146	XPRINT STRETCH	die Größe einer Bitmap auf einer Hostdruckerseite kopieren und ändern	xxx
1147	XPRINT STRETCH PAGE	im Clip- oder Clientbereich der Hostdruckerseite eine Bitmap kopieren und skalieren	xxx
1148	XPRINT STYLE	den Linienstil festlegen, der von verschiedenen XPRINT-Anweisungen verwendet werden soll	xxx
1149	XPRINT TEXT WIDTH	die Größe des Textes berechnen, der auf einem Hostdrucker gedruckt werden soll	xxx
1150	XPRINT WIDTH	die Breite der Grafiklinien festlegen, die von verschiedenen XPRINT-Anweisungen verwendet werden soll	xxx

17. String-Operationen

Die folgenden Funktionen manipulieren und verwalten String-Daten:

LfdNr	Befehl	Beschreibung	Seite
1151	ACODE\$	eine Unicode-Zeichenfolge in eine ANSI - Zeichenfolge konvertieren	xxx
1152	ARRAY ASSIGN	aufeinander folgenden Elementen eines Arrays eine Reihe von Werten zuweisen	xxx
1153	ARRAY DELETE	ein einzelnes Element aus einem bestimmten Array löschen	xxx
1154	ARRAY INSERT	ein einzelnes Element in ein gegebenes Array einfügen	xxx
1155	ARRAY SCAN	ein Array ganz oder teilweise nach einem bestimmten Wert durchsuchen	xxx
1156	ARRAY SORT	das gesamte oder einen Teil eines gegebenen Arrays sortieren	xxx
1157	BIN\$	gibt eine Zeichenfolge mit der binären (Basis 2) Darstellung eines Werts zurück	xxx
1158	BIT\$	den String-Inhalt ohne Änderung kopieren	xxx

1159	BUILD\$	mehrere Zeichenfolgen mit hoher Effizienz verketten	xxx
1160	CHOOSE\$	gibt einen von mehreren Werten zurück, basierend auf dem Wert eines Index.	xxx
1161	CHR\$	einen oder mehrere Zeichencodes in ASCII-Zeichen konvertieren	xxx
1162	CHR\$\$	einen oder mehrere Zeichencodes in Unicode-Zeichen konvertieren	xxx
1163	CHRBYTES	die Größe eines einzelnen Zeichens in einer Zeichenfolgen-Variablen bestimmen	xxx
1164	ChrToOem\$	eine Zeichenfolge aus ANSI/WIDE-Zeichen in OEM-Byte-Zeichen konvertieren	xxx
1165	ChrToUtf8\$	eine Zeichenfolge aus ANSI/WIDE-Zeichen in UTF-8-Byte-Zeichen konvertieren	xxx
1166	CLIP\$	Zeichen aus einer Zeichenfolge löschen	xxx
1167	CLSID\$	gibt eine 16 Byte (128 Bit) lange GUID-Zeichenfolge zurück, die eine CLSID enthält	xxx
1168	COMM LINE	eine mit CR/LF abgeschlossene Zeile von Daten von einem seriellen Port empfangen	xxx
1169	COMM PRINT	eine Zeile binärer Daten über einen seriellen Port senden	xxx
1170	COMM RECV	Binär-Daten von einem seriellen Port empfangen	xxx
1171	COMM SEND	eine Zeichenfolge binärer Daten über einen seriellen Port senden	xxx
1172	COMMAND\$	gibt die zum Starten des Programms verwendete Befehlszeile zurück	xxx
1173	CSET	eine Zeichenfolge innerhalb des Raums einer anderen Zeichenfolge oder eines UDT zentrieren	xxx
1174	CSET\$	gibt eine Zeichenfolge zurück, die eine zentrierte (aufgefüllte) Zeichenfolge enthält	xxx
1175	CURDIR\$	gibt das aktuelle Verzeichnis für ein bestimmtes Laufwerk zurück	xxx
1176	DATA	ein Array von Konstanten deklarieren, das von READ\$ gelesen werden soll	xxx
1177	DATACOUNT	gibt die Gesamtzahl der lokalen Datenelemente zurück	xxx
1178	DATE\$	das Systemdatum festlegen und abrufen	xxx
1179	DEC\$	einen ganzzahligen Wert in eine Dezimal-Zeichenfolge konvergieren	xxx
1180	DIM	deklarieren und dimensionieren von Arrays, Skalarvariablen und Zeiger	xxx
1181	DIR\$	gibt einen Dateinamen zurück, der mit der angegebenen Maske übereinstimmt	xxx
1182	DIR\$ CLOSE	die Freigabe des FindNext-Handles des Betriebssystems erzwingen	xxx
1183	ENVIRON	die Umgebungstabelle des aktuellen Programms ändern	xxx
1184	ENVIRON\$	Zeichenfolgen aus der Umgebungstabelle des Betriebssystems abrufen	xxx
1185	ERASE	Array-Speicher freigeben	xxx

1186	ERL\$	gibt die letzte Bezeichnung, Zeilennummer oder den letzten Prozedurnamen zurück, der vor dem letzten Fehler ausgeführt wurde	xxx
1187	ERROR\$	gibt eine Zeichenfolge zurück, die den beschreibenden Namen eines Fehlers enthält	xxx
1188	EXTRACT\$	bis zum ersten Vorkommen eines angegebenen Zeichens zurückkehren	xxx
1189	EXE	gibt den Pfad und/oder Namen des ausführenden Programms zurück	xxx
1190	FIELD	eine FIELDSTRING-Variable an einen bestimmten Unterabschnitt eines zufälligen Dateipuffers oder einer dynamischen String-Variable binden	xxx
1191	FIELD RESET	die FIELD-Zeichenfolge auf eine dynamische NUL - Zeichenfolge (Länge NULL) zurücksetzen	xxx
1192	FIELD STRING	die FIELD-Zeichenfolge in eine dynamische Zeichenfolge ändern, ihr jedoch zunächst die aktuellen Unterabschnitts-Daten zuweisen	xxx
1193	FILENAME\$	gibt den Dateisystem-Namen einer geöffneten Datei zurück	xxx
1194	FORMAT\$	gibt eine Zeichenfolge mit formatierten numerischen Daten zurück	xxx
1195	FUNCNAME\$	gibt den Namen der aktuellen Subfunktion / Methode / Eigenschaft zurück	xxx
1196	GET	einen Datensatz aus einer Direktzugriffs-Datei lesen	xxx
1197	GET\$	eine Zeichenfolge aus einer im Binärmodus geöffneten Datei lesen	xxx
1198	GET\$\$	WIDE-STRING-Daten aus einer im Binärmodus geöffneten Datei lesen	xxx
1199	GRAPHIC SPLIT	eine Zeichenfolge zur Anzeige auf einem Grafikziel in zwei Teile aufteilen	xxx
1200	GUID\$	gibt eine 16 Byte (128 Bit) lange global eindeutige Kennung (GUID) zurück	xxx
1201	GUIDTXT\$	gibt eine 38 Byte lange, für Menschen lesbare GUID/UUID-Zeichenfolge zurück	xxx
1202	HEX\$	hexadezimale (Basis 16) Zeichenfolgen-Darstellung eines Arguments	xxx
1203	IIF\$	gibt einen von zwei Werten basierend auf einer Wahr/Falsch-Auswertung zurück	xxx
1204	INPUT#	Variable mit Daten aus einer sequentiellen Datei laden	xxx
1205	INPUTBOX\$	eine INPUTBOX\$ zeigt ein Dialogfeld mit einer Eingabeaufforderung an	xxx
1206	INSTR	eine Zeichenfolge durchsuchen nach dem ersten Vorkommen eines Zeichens oder einer Zeichenfolge	xxx
1207	ISNOTNULL	bestimmen, ob eine Zeichenfolge <u>nicht</u> NULL ist (1 oder mehr Zeichen enthält)	xxx
1208	ISNULL	bestimmen, ob eine Zeichenfolge NULL ist (Länge NULL)	xxx
1209	IStringBuilderA.Add	fügt dem Objekt eine ANSI-Zeichenfolge hinzu	xxx

1210	StringBuilderA.Capacity <Get>	die Größe des internen Puffers abrufen	XXX
1211	StringBuilderA.Capacity <Set>	die Größe des internen Puffers festlegen	XXX
1212	StringBuilderA.Char <Get>	gibt den numerischen Zeichencode des Zeichens an der angegebenen Position zurück	XXX
1213	StringBuilderA.Char <Set>	den numerischen Zeichencode des Zeichens an der angegebenen Position ändern	XXX
1214	StringBuilderA.Clear	alle Daten im Objekt werden gelöscht	XXX
1215	StringBuilderA.Delete	löscht eine angegebene Anzahl von Zeichen ab einer angegebenen Position	XXX
1216	StringBuilderA.Insert	eine Zeichenfolge an einer angegebenen Position einfügen	XXX
1217	StringBuilderA.Len	gibt die Anzahl der im Objekt gespeicherten Zeichen zurück	XXX
1218	StringBuilderA.String	der im Objekt gespeicherte ANSI-String wird an den Aufrufer zurückgegeben	XXX
1219	StringBuilderW.Add	dem Objekt eine WIDE-Zeichenfolge anfügen	XXX
1220	StringBuilderW.Capacity <Get>	die Größe des internen Puffers abrufen	XXX
1221	StringBuilderW.Capacity <Set>	die Größe des internen Puffers festlegen	XXX
1222	StringBuilderW.Char <Get>	gibt den numerischen Zeichencode des Zeichens an der angegebenen Position zurück	XXX
1223	StringBuilderW.Char <Set>	den numerischen Zeichencode des Zeichens an der angegebenen Position ändern	XXX
1224	StringBuilderW.Clear	alle Daten im Objekt werden gelöscht	XXX
1225	StringBuilderW.Delete	eine angegebene Anzahl von Zeichen ab einer angegebenen Position werden gelöscht	XXX
1226	StringBuilderW.Insert	eine Zeichenfolge an einer angegebenen Position einfügen	XXX
1227	StringBuilderW.Len	gibt die Anzahl der im Objekt gespeicherten Zeichen zurück	XXX
1228	StringBuilderW.String	der im Objekt gespeicherte WIDE-String wird an den Aufrufer zurückgegeben	XXX
1229	JOIN\$	gibt eine Zeichenfolge zurück, die aus allen Zeichenfolgen in einem Zeichenfolgen-Array besteht	XXX
1230	LCASE\$	gibt die Kleinbuchstaben-Version eines Zeichenfolgen-Arguments zurück	XXX
1231	LEFT\$	gibt die n Zeichen ganz links einer Zeichenfolge zurück	XXX
1232	LEN	gibt die logische Länge einer Variablen, eines UDT oder einer UNION zurück	XXX
1233	LET	einer Variablen einen Wert zuweisen	XXX
1234	LET (with Types)	einer benutzerdefinierten Typvariablen Daten zuweisen	XXX
1235	LET (with Variants)	einer Variant-Variablen einen Wert oder eine Objektreferenz zuweisen	XXX
1236	LINE INPUT#	aus einer sequenziellen Datei Zeile(n) in eine Zeichenfolgen-Variable oder ein Array einlesen	XXX
1237	LPRINT	Text und Daten an ein Druckergerät ausgeben	XXX
1238	LPRINT\$	gibt das aktuelle Druckergerät zurück, das für LPRINT-Vorgänge verwendet wird	XXX

1239	LSET	eine Zeichenfolge innerhalb des Bereichs einer anderen Zeichenfolge oder eines UDT linksbündig ausrichten	xxx
1240	LSET\$	gibt eine Zeichenfolge zurück, die eine linksbündige (aufgefüllte) Zeichenfolge enthält	xxx
1241	LTRIM\$	gibt eine Zeichenfolge zurück, aus der führende Zeichen oder Zeichenfolgen entfernt wurden	xxx
1242	MAX\$	gibt das Argument mit dem größten (maximalen) Wert zurück	xxx
1243	MCASE\$	gibt eine Version eines Zeichenfolgen-Arguments in gemischter Groß- und Kleinschreibung zurück	xxx
1244	MID\$	gibt einen Teil einer Zeichenfolge zurück	xxx
1245	MID\$	Zeichen in einer Zeichenfolge durch Zeichen aus einer anderen Zeichenfolge ersetzen	xxx
1246	MIN\$	gibt das Argument mit dem kleinsten (minimalen) Wert zurück	xxx
1247	MKBYT\$	einen Byte-Wert in eine binär codierte Zeichenfolge konvertieren	xxx
1248	MKCUR\$	einen Währungswert in eine binär codierte Zeichenfolge konvertieren	xxx
1249	MKCUX\$	einen erweiterten Währungswert in eine binär codierte Zeichenfolge konvertieren	xxx
1250	MKD\$	einen Wert mit doppelter Genauigkeit in eine binär codierte Zeichenfolge konvertieren	xxx
1251	MKDWD\$	einen DoubleWord-Wert in eine binär codierte Zeichenfolge konvertieren	xxx
1252	MKE\$	einen Wert mit erweiterter Genauigkeit in eine binär codierte Zeichenfolge konvertieren	xxx
1253	MKI\$	einen ganzzahligen Wert in eine binär codierte Zeichenfolge konvertieren	xxx
1254	MKL\$	einen LongInteger-Wert in eine binär codierte Zeichenfolge konvertieren	xxx
1255	MKQ\$	einen QuadInteger-Wert in eine binär codierte Zeichenfolge konvertieren	xxx
1256	MKS\$	einen Wert mit einfacher Genauigkeit in eine binär codierte Zeichenfolge konvertieren	xxx
1257	MKWRD\$	einen word-Wert in eine binär codierte Zeichenfolge konvertieren	xxx
1258	MKDIR	erstellt ein Unterverzeichnis/einen Ordner (wie mit dem DOS-Befehl MKDIR)	xxx
1259	NUL\$	gibt eine Zeichenfolge zurück, die eine angegebene Anzahl von \$NUL-Zeichen enthält	xxx
1260	OBJRESULT\$	gibt eine Zeichenfolge zurück, die einen OBJRESULT-Code (hResult) beschreibt	xxx
1261	OCT\$	gibt eine Zeichenfolge zurück, die eine oktale (Basis 8) Darstellung eines Werts ist	xxx
1262	OemToChr\$	eine Bytefolge von OEM-Zeichen in ANSI/WIDE-Zeichen konvertieren	xxx

1263	PARSE	eine Zeichenfolge analysieren und alle durch Trennzeichen getrennten Felder in ein Array extrahieren	xxx
1264	PARSE\$	gibt ein durch Trennzeichen getrenntes Feld aus einem Zeichenfolgen-Ausdruck zurück	xxx
1265	PARSECOUNT	gibt die Anzahl der durch Trennzeichen getrennten Felder in einem Zeichenfolgen-Ausdruck zurück	xxx
1266	PATHNAME\$	einen Pfad-/Dateinamen analysieren, um Komponententeile zu extrahieren	xxx
1267	PATHSCAN\$	eine Datei auf der Festplatte suchen und den Pfad und/oder Teile des Dateinamens zurückgeben	xxx
1268	PEEK\$	gibt aufeinanderfolgende 1-Byte-Zeichen zurück, beginnend an einer bestimmten Speicherposition	xxx
1269	PEEK\$\$	gibt aufeinanderfolgende 2-Byte-Zeichen zurück, beginnend an einem bestimmten Speicherort	xxx
1270	POKE\$	eine Bytefolge speichern, beginnend an einem bestimmten Speicherort	xxx
1271	POKE\$\$	eine Sequenz von 2-Byte-Zeichen speichern, beginnend an einem bestimmten Speicherort	xxx
1272	PRINT#	ein vollständiges Array in eine sequentielle Datei schreiben	xxx
1273	PROGID\$	gibt die alphanumerische PROGID-Zeichenfolge (Text) einer bestimmten CLSID zurück	xxx
1274	PUT	einen Datensatz in eine Direktzugriffsdatei oder eine Variable in eine Binärdatei schreiben	xxx
1275	PUT\$	einen ANSI-String in eine im Binärmodus geöffnete Datei schreiben	xxx
1276	PUT\$\$	eine WIDE Unicode-Zeichenfolge in eine im Binärmodus geöffnete Datei schreiben	xxx
1277	READ\$	Zeichenfolgen-Daten aus einer lokalen DATA-Liste abrufen	xxx
1278	REGEXPR	eine Zeichenfolge nach einem passenden Platzhalter oder regulären Ausdruck durchsuchen	xxx
1279	REGREPL	nach einer "Platzhalter"-Übereinstimmung in einer Zeichenfolge scannen mit einer neuen Zeichenfolge	xxx
1280	REMAIN\$	gibt den Teil einer Zeichenfolge zurück, der auf das erste Vorkommen eines Zeichens oder einer Zeichengruppe folgt	xxx
1281	REMOVE\$	gibt eine Kopie einer Zeichenfolge zurück, aus der Zeichen oder Zeichenfolgen entfernt wurden	xxx
1282	REPEAT\$	gibt eine Zeichenfolge zurück, die aus mehreren Kopien einer angegebenen Zeichenfolge besteht	xxx
1283	REPLACE	alle Vorkommen einer Zeichenfolge durch eine andere Zeichenfolge ersetzen	xxx
1284	RESET	eine Zeichenfolge, einen Zeichenfolgenarray-Index oder ein ganzes Array löschen	xxx
1285	RESOURCE\$	gibt die vordefinierten Ressourcen-Daten zurück	xxx
1286	RETAIN\$	gibt eine Zeichenfolge zurück, aus der alle nicht angegebenen Zeichen entfernt wurden	xxx
1287	RIGHT\$	gibt die n Zeichen ganz rechts einer Zeichenfolge zurück	xxx

1288	RSET	eine Zeichenfolge rechtsbündig im Bereich einer Zeichenfolgen-Variablen oder eines UDT ausrichten	XXX
1289	RSET\$	gibt eine Zeichenfolge zurück, die eine rechtsbündige (aufgefüllte) Zeichenfolge enthält	XXX
1290	RTRIM\$	gibt eine Kopie einer Zeichenfolge zurück, aus der die nachfolgenden Zeichen/Zeichenfolgen entfernt wurden	XXX
1291	SHRINK\$	eine Zeichenfolge verkürzen, um ein konsistentes Einzelzeichen als Trennzeichen zu verwenden	XXX
1292	SIZEOF	gibt die Gesamtlänge oder physikalische Länge einer beliebigen <i>PowerBASIC</i> -Variable zurück	XXX
1293	SPACE\$	gibt eine Zeichenfolge zurück, die aus einer bestimmten Anzahl von Leerzeichen besteht	XXX
1294	SPLIT	eine Zeichenfolge in zwei Teile aufteilen	XXX
1295	STR\$	gibt die Zeichenfolgen-Darstellung einer Zahl in druckbarer Form zurück	XXX
1296	STRDELETE\$	eine angegebene Anzahl Zeichen aus einem Zeichenfolgen-Ausdruck löschen	XXX
1297	STRING\$	gibt eine ANSI-Zeichenfolge zurück, die aus mehreren Kopien eines angegebenen Zeichens besteht	XXX
1298	STRING\$\$	gibt eine WIDE-Zeichenfolge zurück, die aus mehreren Kopien eines angegebenen Zeichens besteht	XXX
1299	STRINSERT\$	eine Zeichenfolge an einer bestimmten Position innerhalb einer anderen Zeichenfolge einfügen	XXX
1300	STRPTR	gibt die Adresse der Daten zurück, die in einer Zeichenfolge variabler Länge enthalten sind	XXX
1301	STRREVERSE\$	den Inhalt eines Zeichenfolgen-Ausdrucks umkehren	XXX
1302	SWAP	die Werte zweier Zeichenfolgen, Zeiger oder Zeigerziele austauschen	XXX
1303	SWITCH\$	gibt ein Element einer Reihe basierend auf einer Wahr/Falsch-Auswertung zurück	XXX
1304	TAB\$	gibt eine Zeichenfolge mit TAB-Zeichen zurück, die durch Leerzeichen erweitert wurden	XXX
1305	TALLY	die Anzahl der Vorkommen angegebener Zeichen/Zeichenfolgen zählen	XXX
1306	TIME\$	die Systemzeit lesen und / oder einstellen	XXX
1307	TRIM\$	gibt eine Zeichenfolge zurück, bei der führende und nachfolgende Zeichen entfernt wurden	XXX
1308	TYPE SET	einem UDT den Wert eines UDT oder Zeichenfolgen-Ausdrucks zuweisen	XXX
1309	UCASE\$	gibt eine vollständig in Großbuchstaben geschriebene Version einer Zeichenfolge zurück	XXX
1310	UCODE\$	eine ANSI-Zeichenfolge in eine Unicode-Zeichenfolge konvertieren	XXX
1311	UCODEPAGE	die Standard-Codepage festlegen, die für ANSI/UNICODE-Konvertierungen verwendet wird	XXX

1312	UNWRAP\$	gepaarte Zeichen vom Anfang und Ende einer Zeichenfolge entfernen	xxx
1313	USING\$	Zeichenfolgen-/numerische Ausdrücke mithilfe einer Masken-Zeichenfolge formatieren	xxx
1314	Utf8ToChr\$	eine Bytefolge von OEM-Zeichen in ANSI/WIDE-Zeichen konvertieren	xxx
1315	VAL function	gibt das numerische Äquivalent eines Zeichenfolgen-Arguments zurück	xxx
1316	VAL statement	eine Textzeichenfolge in einen numerischen Wert mit zusätzlichen Informationen konvertieren	xxx
1317	VARIANT\$	gibt den in einer Variant-Variablen enthaltenen dynamischen ANSI-String zurück	xxx
1318	VARIANT\$\$	gibt die dynamische Unicode-Zeichenfolge zurück, die in einer Variant-Variablen enthalten ist	xxx
1319	VARPTR	gibt die 32-Bit-Adresse eines String-Handles zurück	xxx
1320	VERIFY	bestimmen, ob jedes Zeichen einer Zeichenfolge in einer anderen Zeichenfolge enthalten ist	xxx
1321	WRAP\$	am Anfang und Ende einer Zeichenfolge gepaarte Zeichen hinzufügen	xxx

18. Textbefehle

Die folgenden Befehle können mit einem Textfenster verwendet werden:

LfdNr	Befehl	Beschreibung	Seite
1322	TXT.CELL	die Cursorposition festlegen oder sie abrufen	xxx
1323	TXT.CLS	das Textfenster löschen und den Cursor in die obere linke Ecke verschieben	xxx
1324	TXT.COLOR	die Vordergrundfarbe festlegen	xxx
1325	TXT.END	das aktuell am Programm angehängte Textfenster wird zerstört und vom Prozess getrennt	xxx
1326	TXT.INKEY\$	ein Tastaturzeichen lesen, wenn eines bereit ist	xxx
1327	TXT.INSTAT	bestimmt, ob ein Tastaturzeichen bereit ist	xxx
1328	TXT.LINE.INPUT	eine ganze Zeile von der Tastatur lesen	xxx
1329	TXT.PRINT	Textdaten in das TEXT WINDOW an der aktuellen Cursorposition schreiben	xxx
1330	TXT.WAITKEY\$	Tastaturzeichen einlesen und warten, bis eines bereit ist	xxx
1331	TXT.WINDOW	ein neues Textfenster wird erstellt und an das Programm angehängt	xxx

19. Thread-Steuerung

Die folgenden Funktionen dienen zum Erstellen und Verwalten von Threads:

LfdNr	Befehl	Beschreibung	Seite
1332	IPowerThread.Close	gibt den Thread-Handle dieses Threads frei	xxx
1333	IPowerThread.Equals	das angegebene Objekt vergleichen, um festzustellen, ob es auf dasselbe Objekt wie dieses Objekt verweist	xxx
1334	IPowerThread.Handle	das Handle des Threads zur Verwendung mit Windows-API-Funktionen abrufen	xxx
1335	IPowerThread.Id	die ID des Threads zur Verwendung mit Windows-API-Funktionen abrufen	xxx
1336	IPowerThread.IsAlive	den Thread überprüfen, um festzustellen, ob er derzeit aktiv ist	xxx
1337	IPowerThread.Join	warten bis das angegebene Thread-Objekt abgeschlossen ist, bevor die Ausführung dieses Threads fortgesetzt wird	xxx
1338	IPowerThread.Launch	beginnt mit der Ausführung des Thread-Objekts	xxx
1339	IPowerThread.Priority <Get>	den Prioritätswert für diesen Thread abrufen	xxx
1340	IPowerThread.Priority <Set>	den Prioritätswert für diesen Thread festlegen	xxx
1341	IPowerThread.Result	wenn der Thread beendet wurde, wird der Ergebniswert abgerufen und an den Aufrufer zurückgegeben	xxx
1342	IPowerThread.Resume	die Ausführung eines angehaltenen Threads fortsetzen	xxx
1343	IPowerThread.StackSize <Get>	die Größe des Stapels für diesen Thread abrufen	xxx
1344	IPowerThread.StackSize <Set>	die Größe des Stapels für diesen Thread auf den angegebenen Wert setzen	xxx
1345	IPowerThread.Suspend	die Ausführung des Threads unterbrechen	xxx
1346	IPowerThread.TimeCreate	das Datum und die Uhrzeit der Thread-Erstellung abrufen	xxx
1347	IPowerThread.TimeExit	das Datum und die Uhrzeit des Thread-Exits abrufen	xxx
1348	IPowerThread.TimeKernel	die Zeit abrufen, die dieser Thread im Kernelmodus verbracht hat	xxx
1349	IPowerThread.TimeUser	die Zeit abrufen, die dieser Thread im Benutzer-Modus verbracht hat	xxx
1350	PROCESS GET PRIORITY	den Prioritätswert für den aktuellen Prozess abrufen	xxx
1351	PROCESS SET PRIORITY	den Prioritätswert für den aktuellen Prozess festlegen	xxx
1352	THREADED	die Thread Local Storage (TLS)-Variablen deklarieren	xxx
1353	THREAD CLOSE	einen Windows-Thread schließen	xxx
1354	THREAD CREATE	einen Windows-Thread erstellen	xxx
1355	THREAD GET PRIORITY	den Prioritätswert für einen Thread abrufen	xxx
1356	THREAD FUNCTION	eine Thread-Funktion deklarieren	xxx
1357	THREAD SET PRIORITY	den Prioritätswert für einen Thread festlegen	xxx

1358	THREAD RESUME	die Ausführung eines angehaltenen Windows-Threads fortsetzen	xxx
1359	THREAD STATUS	den Status eines Windows-Threads abrufen	xxx
1360	THREAD SUSPEND	die Ausführung eines Windows-Threads unterbrechen	
1361	THREADCOUNT	gibt die Anzahl der aktiven Threads zurück, die in einem Modul vorhanden sind	xxx
1362	THREADID	gibt eine Long-Integer-Thread-Kennung des aktuellen Threads zurück	xxx

20. Zeitbefehle

Die folgenden Funktionen manipulieren und verwalten die Uhrzeit und das Systemdatum:

LfdNr	Befehl	Beschreibung	Seite
1363	DATE\$	das Systemdatum festlegen oder abrufen	xxx
1364	DAYNAME\$	eine Wochentagszahl in den zugehörigen Namen konvertieren	xxx
1365	IPowerTime.AddDays	eine angegebene Anzahl von Tagen zum Wert dieses Objekts addieren oder subtrahieren	xxx
1366	IPowerTime.AddHours	eine angegebene Anzahl von Stunden zum Wert dieses Objekts addieren oder subtrahieren	xxx
1367	IPowerTime.AddMinutes	eine angegebene Anzahl von Minuten zum Wert dieses Objekts addieren oder subtrahieren	xxx
1368	IPowerTime.AddMonths	eine angegebene Anzahl von Monaten zum Wert dieses Objekts addieren oder subtrahieren	xxx
1369	IPowerTime.AddMSeconds	eine angegebene Anzahl von Millisekunden zum Wert dieses Objekts addieren oder subtrahieren	xxx
1370	IPowerTime.AddSeconds	eine angegebene Anzahl von Sekunden zum Wert dieses Objekts addieren oder subtrahieren	xxx
1371	IPowerTime.AddTicks	eine angegebene Anzahl von Ticks zum Wert dieses Objekts addieren oder subtrahieren	xxx
1372	IPowerTime.AddYears	eine angegebene Anzahl von Jahren zum Wert dieses Objekts addieren oder subtrahieren	xxx
1373	IPowerTime.DateDiff	die Datumskomponente eines externen PowerTime-Objekts mit der Datumskomponente dieses Objekts vergleichen	xxx
1374	IPowerTime.DateString	gibt die Datumskomponente des Objekts als Zeichenfolge zurück	xxx
1375	IPowerTime.DateStringLong	gibt die Datumskomponente des PowerTime-Objekts als Zeichenfolge mit vollständigen alphabetischen Monatsnamen zurück	xxx
1376	IPowerTime.Day	gibt die Tageskomponente des Objekts zurück	xxx
1377	IPowerTime.DayOfWeek	gibt die Wochentagskomponente des Objekts zurück	xxx
1378	IPowerTime.DayOfWeekString	gibt den Wochentag des Objekts als Zeichenfolge (Sonntag, Montag ...) zurück	xxx

1379	IPowerTime.DaysInMonth	gibt die Anzahl der Tage zurück, die den Monat des Datums des PowerTime-Objekts umfassen	xxx
1380	IPowerTime.FileTime <Get>	gibt einen Quad-Integer-Wert des PowerTime-Objekts als FileTime zurück	xxx
1381	IPowerTime.FileTime <Set>	der durch den Parameter angegebene FileTime Quad-Integer-Wert wird als PowerTime-Objektwert zugewiesen	xxx
1382	IPowerTime.Hour	gibt die Stundenkomponente des Objekts zurück	xxx
1383	IPowerTime.IsLeapYear	gibt "true" / "false" (-1 / 0) zurück, um anzugeben, ob das Objektjahr ein Schaltjahr ist	xxx
1384	IPowerTime.Minute	gibt die Minuten-Komponente des Objekts zurück	xxx
1385	IPowerTime.Month	gibt die Monats-Komponente des Objekts zurück	xxx
1386	IPowerTime.MonthString	gibt die Monats-Komponente des Objekts als Zeichenfolge (Januar, Februar ...) zurück	xxx
1387	IPowerTime.MSecond	gibt die ms-Komponente des PowerTime-Objekts zurück	xxx
1388	IPowerTime.NewDate	der Datums-Komponente des PowerTime-Objekts einen neuen Wert zuweisen	xxx
1389	IPowerTime.NewTime	der Zeit-Komponente des PowerTime-Objekts einen neuen Wert zuweisen	xxx
1390	IPowerTime.Now	dem Objekt wird das aktuelle lokale Datum und die Uhrzeit auf diesem Computer zugewiesen	xxx
1391	IPowerTime.NowUTC	dem Objekt wird das aktuelle koordinierte Weltdatum und die aktuelle koordinierte Weltzeit (UTC) zugewiesen	xxx
1392	IPowerTime.Second	gibt die zweite Komponente des Objekts zurück	xxx
1393	IPowerTime.Tick	gibt die Tick-Komponente des Objekts zurück	xxx
1394	IPowerTime.TimeDiff	vergleicht die Zeitkomponente eines externen PowerTime-Objekts mit der Zeitkomponente dieses Objekts	xxx
1395	IPowerTime.TimeString	gibt die Zeit-Komponente des PowerTime-Objekts als Zeichenfolge zurück	xxx
1396	IPowerTime.TimeString24	gibt die Zeit-Komponente des PowerTime-Objekts als Zeichenfolge zurück, formatiert im 24-Stunden-Format als hh:mm	xxx
1397	IPowerTime.TimeStringFull	gibt die Zeitkomponente des PowerTime-Objekts als Zeichenfolge zurück, formatiert im 24-Stunden-Format als hh:mm:ss.tt	xxx
1398	IPowerTime.Today	das aktuelle lokale Datum des Computers wird diesem PowerTime-Objekt zugewiesen, was für Anwendungen mit Datumsangaben geeignet ist	xxx
1399	IPowerTime.ToLocalTime	das Objekt wird in die Ortszeit konvertiert	xxx
1400	IPowerTime.ToUTC	das Objekt wird in die Koordinierte Weltzeit (UTC) konvertiert	xxx
1401	IPowerTime.Year	gibt die Jahres-Komponente des PowerTime-Objekts als numerischen Wert zurück	xxx
1402	MONTHNAME\$	eine Monats-Zahl wird in den zugehörigen Namen konvertiert	xxx

1403	SLEEP	der aktuelle Thread pausiert für eine angegebene Anzahl von Millisekunden	xxx
1404	TIME\$	die Systemzeit lesen und / oder einstellen	xxx
1405	TIMER	gibt die Anzahl der Sekunden zurück, die seit Mitternacht vergangen sind	xxx
1405	TIX	die verstrichenen CPU-Zyklen messen	xxx

21. Sonstige Operationen

Sonstige Funktionen:

ASM: eine Assembler-Anweisung identifizieren

LfdNr	Befehl	Beschreibung	Seite
1406	ASM ALIGN	den Befehlsort auf eine Zweierpotenz-Adresse aufrunden	xxx
1407	ASMDATA/END ASMDATA	einen Block definieren, in dem primitive schreibgeschützte Daten gespeichert werden	xxx
1408	BEEP	den Windows-Standardton über die Computer-Lautsprecher wiedergeben	xxx
1409	IMPORT	eine Bibliothek (DLL) laden oder freigeben, um auf eine importierte Prozedur zu zugreifen	xxx
1410	PLAY WAVE	einen Ton unter Programmsteuerung abspielen	xxx
1411	PLAY WAVE END	alle aktuell abgespielten Wellenform-Töne stoppen	xxx
1412	REM	gibt an, dass der Rest einer Quellcodezeile eine Anmerkung oder ein Kommentar ist	xxx